

**Tahdonalaisen lihasaktiviteetin havaitseminen  
EMG-signaalista neuroverkon avulla**

Pekka-Henrik Niemenlehto

Tampereen yliopisto  
Tietojenkäsittelytieteiden laitos  
Pro gradu -tutkielma  
Joulukuu 2004

Tampereen yliopisto  
Tietojenkäsittelytieteiden laitos  
Pekka-Henrik Niemenlehto: Tahdonalaisen lihasaktiviteetin havaitseminen  
EMG-signaalista neuroverkon avulla  
Pro gradu -tutkielma, 70 sivua  
Joulukuu 2004

---

Tutkielmassa tarkastellaan, kuinka neuroverkot soveltuvat lihaksen sähköisen toiminnan muutosten havaitsemiseen. Erityisesti käsitellään muutoksia, jotka kertovat tahdonalaisista lihastoiminnoista. Muutosten havaitseminen tapahtuu lihaksen sähköistä toimintaa kuvaavan signaalin perusteella. Tehtävää varten suunniteltiin neuroverkkopohjainen ilmaisinjärjestelmä, jota testattiin kokeellisesti. Tässä vaiheessa keskityttiin monikerrospereptron-neuroverkkoihin ja järjestelmän toimintaan ei-reaaliaikaisessa ympäristössä. Lisäksi tarkasteltiin sitä, kuinka digitaalisten esikäsittelymenetelmien soveltaminen vaikuttaa neuroverkkojen suorituskykyyn. Tutkimusaineisto muodostui kolmenkymmenen kohenkilön kasvolihasen toimintaa kuvaavista signaaleista, jotka mitattiin elektromyografia-tekniikalla. Signaaleista valittiin kolmen tyyppisiä signaali-ikkunoita, joita luokiteltiin neuroverkoilla vastaaviin luokkiinsa: lihasaktiviteetin alkaminen, lihasaktiviteetin päättymisen ja tasannekohta, jonka aikana ei ole havaittavissa merkittäviä lihaksen sähköisen toiminnan muutoksia. Monikerrospereptron-neuroverkon piiloneuronien lukumäärällä ei havaittu olevan kovinkaan suurta vaikutusta järjestelmän suorituskykyyn. Sen sijaan esikäsittelymenetelmän valinnalla havaittiin olevan suurempi vaikutus, kun signaalien kohinatasoa kasvatettiin keinotekoisesti. Parhaiten vaihtelevissa kohinaolosuhteissa toimi aalloke-kohinanpoisto. Kohinatasosta riippuen sen avulla saavutettiin 93–97 %:n luokittelutarkkuus.

CR-luokat: I.5.1 **[Pattern Recognition]**: Models - *Neural nets*; I.5.4 **[Pattern Recognition]**: Applications - *Signal processing*.

Avainsanat ja -sanonnat: digitaalinen signaalinkäsittely, elektromyografia, hahmontunnistus, neuroverkot.

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
<b>2</b>	<b>Digitaalinen signaalinkäsittely</b>	<b>3</b>
2.1	Diskreettiaikaiset signaalit . . . . .	3
2.2	Diskreettiaikaiset järjestelmät . . . . .	5
2.3	Diskreetti Fourier-muunnos . . . . .	8
2.4	Aallope-muunnos . . . . .	11
<b>3</b>	<b>Neuroverkot</b>	<b>16</b>
3.1	Neuronit . . . . .	17
3.2	Neuroverkkoarkkitehtuureja . . . . .	19
3.3	Neuroverkon oppiminen . . . . .	21
3.4	Neuroverkkojen sovelluksia . . . . .	23
3.5	Monikerrosperceptron-neuroverkot . . . . .	24
<b>4</b>	<b>Elektromyogrammi</b>	<b>28</b>
4.1	Lihasten rakenne ja toiminta . . . . .	28
4.2	Sähköisen lihasaktiiviteetin mittaaminen . . . . .	30
4.3	EMG:n kvantifiointi . . . . .	33
4.4	EMG:n sovelluksia . . . . .	35
<b>5</b>	<b>Neuroverkkopohjainen ilmaisinjärjestelmä</b>	<b>37</b>
5.1	Ilmaisinjärjestelmän suunnitelma . . . . .	37
5.2	Tutkimusaineiston kerääminen . . . . .	43
5.3	Testien toteutus . . . . .	48
5.4	Tulosten analysointia . . . . .	53
<b>6</b>	<b>Yhteenveto</b>	<b>63</b>
	<b>Lähdeluettelo</b>	<b>67</b>

# Luku 1

## Johdanto

Sopivien mittalaitteiden avulla ihmiskehosta voidaan mitata erilaisia biosignaaleja. Niiden avulla saadaan informaatiota elimistön tilasta ja toiminnasta. Eniten biosignaaleja hyödyntävä tieteenala on lääketiede ja tunnetuin biosignaalien sovelluskohde lieneekin lääketieteellinen diagnoosi. Sen tavoitteena on elimistön toimintaan vaikuttavien tekijöiden, kuten tautien ja vaurioiden, tunnistaminen. Biosignaalien sovelluskohteet eivät rajoitu yksinään lääketieteelliseen diagnoosiin, vaan niitä voidaan käyttää myös toisenlaisissa sovelluksissa.

Elektromyogrammi on lihaksen sähköistä toimintaa kuvaava biosignaali. Lääketieteessä sitä sovelletaan lihastutkimuksessa ja sen avulla voidaankin diagnosoida eri tyyppisiä lihassairauksia. Elektromyogrammia on käytetty myös syötekanavana käyttöliittymäsovelluksissa, kuten tietokoneen, pyörätuolin ja keinotekoisien proteesien ohjauksessa. Käyttöliittymäsovelluksissa biosignaalista pyritään havaitsemaan sellaisia ilmiöitä, jotka viestivät tietynlaisista fysiologisista tapahtumista. Elektromyogrammin kohdalla niitä ovat motoriset tapahtumat; erityisesti lihasaktiviteetin alkaminen ja päättyminen.

Motoristen tapahtumien havaitsemiseen voidaan soveltaa useita signaalinkäsittelyn menetelmiä. Tavallisesti menetelmät ovat parametrisia ja niiden soveltaminen edellyttääkin parametrien määrittämisen tapauskohtaisesti. Ilman oikeanlaisia parametrien arvoja ei voida taata menetelmien virheetöntä toimintaa. Sopivien parametrien määrittäminen voi olla vaikeaa tai työlästä. Kyseessä saattaa olla myös ajallisesti vaativa tehtävä. Yleistämiskykyinen menetelmä olisi ihanteellinen, koska se kykenisi jo perustilassaan suoriutumaan tehtävästä useimmissa tapauksissa. Parametrien määrittäminen olisi keskimäärin helpompaa ja riittävän suorituskyvyn saavuttamiseksi se olisi useimmiten tarpeetonta.

Neuroverkot ovat hajautettuja rinnakkaisprosessoivia järjestelmiä, joilla pyritään mallintamaan joitain ihmisaivojen toimintoja. Yksinkertaisesti ne voidaan kuvata laskentajärjestelminä, jotka käsittelevät saamansa syötteen jollain erityisellä tavalla ja tuottavat sitten tuloksen. Ihmisaivojen tavoin neuroverkot oppivat

toimintansa käsiteltävän informaation perusteella. Lisäksi neuroverkot kykenevät yleistämään opittua informaatiota ihmisaivojen tavoin, minkä vuoksi ne voivat reagoida mielekkäästi myös sellaisiin syötteisiin, joita ne eivät ole aikaisemmin kohdanneet.

Tutkielmassa tarkastellaan, kuinka neuroverkot soveltuvat lihaksen sähköisen toiminnan muutosten havaitsemiseen. Erityisesti käsitellään muutoksia, jotka kertovat tahdonalaisista lihastoiminnoista. Muutosten havaitseminen tapahtuu lihaksen sähköistä toimintaa kuvaavan signaalin perusteella. Tehtävää varten suunniteltiin neuroverkkopohjainen ilmaisinjärjestelmä, jonka toimintaa testattiin kokeellisesti. Tässä vaiheessa keskityttiin monikerrosperceptron-neuroverkkoihin ja järjestelmän toimintaan ei-reaaliaikaisessa ympäristössä. Lisäksi tarkasteltiin sitä, kuinka digitaalisten esikäsittelymenetelmien soveltaminen vaikuttaa neuroverkkojen suorituskyykyyn vaihtelevissa kohinaolosuhteissa.

Tutkielma voidaan jakaa kahteen osaan. Aluksi käsitellään tutkielman kannalta olennaisia teoreettisia aiheita. Niitä ovat digitaalinen signaalinkäsittely (luku 2), neuroverkot (luku 3) ja elektromyografia (luku 4). Teoriaosuuden jälkeen perehdytään varsinaiseen tutkimukseen. Luvussa 5 esitellään suunniteltu ilmaisinjärjestelmä, käsitellään tutkimusaineiston keräämistä ja ilmaisinjärjestelmän testaamista, sekä analysoidaan testien tuloksia. Lopuksi luvussa 6 muodostetaan tutkielman yhteenveto, pohditaan tutkimusasetelman kehittämistä ja esitellään mahdollisten jatkotutkimusten aiheita.

# Luku 2

## Digitaalinen signaalinkäsittely

Signaali määritellään tavallisesti muuttujaksi, funktioksi tai ilmiöksi, joka siirtää informaatiota. Signaalien siirtämän informaation johdosta ne ovatkin tärkeässä asemassa useissa tekniikan ja tieteen sovelluksissa. Esimerkiksi lääketieteen, tietotekniikan, viestintätekniikan tai audiovisuaalisen alan sovelluksissa erilaiset signaalit ovat keskeisessä asemassa. Digitaalisia signaaleja ja niiden käsittelyä tutkivaa tieteenalaa kutsutaan *digitaaliseksi signaalinkäsittelyksi* (digital signal processing, DSP). Tavallisesti signaalinkäsittelyn tavoitteena on häiriön poistaminen signaalista, signaalin muuntaminen sovelluksen kannalta sopivampaan muotoon tai signaalin analysointi jollain hyödyllisellä tavalla. Tässä luvussa luodaan katsaus tutkielman kannalta tärkeisiin DSP:n aiheisiin ja käsitellään niiden ymmärtämisen kannalta olennaisia perustietoja.

### 2.1 Diskreettiaikaiset signaalit

Mittauksen tuloksena muodostuva signaali on useimmiten analoginen. Analogiset signaalit ovat jatkuva-aikaisia, jolloin ne on määritelty kaikilla ajanhetkillä. Lisäksi analogiset signaalit voivat saada arvoja äärettömältä lukuväliltä. Näiden syiden johdosta analogisten signaalien käsittelyyn ei voida suoranaisesti soveltaa DSP:n menetelmiä. Analoginen signaali voidaan kuitenkin muuntaa digitaaliseksi, minkä jälkeen DSP:n soveltaminen on mahdollista. Digitaaliset signaalit voidaan muuntaa myös (takaisin) analogiseen muotoon. Nykyään suuri osa signaalinkäsittelystä tehdään pääasiassa digitaalisilla laitteilla. Ifeachorin ja Jervisin [24] mukaan DSP:tä käytetään myös sellaisissa sovelluksissa, jotka ovat vaikeita tai mahdottomia toteuttaa analogisilla menetelmillä.

*Näytteistyksellä* (sampling) tarkoitetaan diskreettiaikaisen signaalin muodostamista jatkuva-aikaisesta signaalista. Sen tuloksena saadaan signaali, joka kuvaa alkuperäistä jatkuva-aikaista signaalia vain tiettyinä ajanhetkinä. Diskreettiaikai-

nen signaali määritellään kaksisuuntaisena lukujonona

$$x(n) = x_c(nT),$$

missä  $n$  on kokonaisluku ja  $x_c(nT)$  on näytteistettävä jatkuva-aikainen signaali. Vakiota  $T$  kutsutaan *näytteenottoväliksi* (sampling interval). Se on näytearvojen välinen aika sekunneissa. Lukujonon yleistä termiä  $x(n)$  kutsutaan diskreettiaikaisen signaalin  $n$ :neksi *näytearvoksi* (sample).

*Näytteenottotaajuus* (sampling frequency, sampling rate) kertoo, kuinka usein jatkuva-aikaisesta signaalista otetaan näytearvoja. Se määritellään näytteenottovälin käänteislukuna:

$$F_s = 1/T.$$

Näytteenottotaajuuden yksikkö on hertsi (Hz). Jatkuva-aikaisia signaaleja näytteistettäessä näytteenottotaajuuden tulee olla vähintään kaksi kertaa niin suuri kuin signaalissa esiintyvä korkein taajuuskomponentti. Sääntöä kutsutaan *näytteenottoteoreemaksi* (sampling theorem). Mikäli jatkuva-aikainen signaali näytteistetään matalammalla näytteenottotaajuudella, niin silloin tapahtuu laskostumista.

*Laskostumisella* (aliasing) tarkoitetaan korkeiden taajuuksien kuvautumista matalammiksi taajuuksiksi, mikä estää alkuperäisen signaalin muodostamisen näytearvojen perusteella. Käytännössä laskostumisen aiheuttaa näytteistettävän signaalin taajuusinformaatio, joka sijaitsee *Nyquistin taajuutta* (Nyquist frequency) korkeammalla taajuusalueella. Nyquistin taajuus on puolet näytteenottotaajuudesta. Laskostuminen pyritään välttämään käyttämällä riittävän korkeaa näytteenottotaajuutta ja *laskostumisenestosuodinta* (anti-aliasing filter). Suodatuksen tavoitteena on poistaa Nyquistin taajuuden ylittävä taajuusinformaatio signaalista ennen näytteistystä.

Näytteistyksen tuloksena saadaan jatkuva-aikaisen signaalin diskreettiaikainen esitys. DSP:n kannalta tämä ei yksinään riitä, sillä myös signaalin arvo- eli amplitudiasteikko on muunnettava diskreetiksi. *Kvantisoinnin* (quantization) tarkoituksena on koodata jatkuvan asteikon näytearvot diskreetille asteikolle. Diskreetti asteikko koostuu äärellisestä määrästä arvoja, ja sen tarkkuus määräytyy kvantisoinnissa käytettävien bittien lukumäärän mukaan. Analogisen signaalin näytteistyksen ja näytearvojen kvantisoinnin tuloksena on varsinainen digitaalinen signaali. Kokonaisprosessia kutsutaan *analogia-digitaali-muunnokseksi* (analog-to-digital conversion) tai lyhyemmin A/D-muunnokseksi. *Digitaali-analogia-muunnos* (digital-to-analog conversion) tai D/A-muunnos on sen käänteinen operaatio. Tässä esitetty kuvaus A/D-muunnoksesta on tutkielman kannalta riittävä, mutta niin A/D- kuin D/A-muunnostakin käsitellään yksityiskohtaisemmin kirjallisuudessa (katso esim. [24] ja [40]).

Diskreettiaikaisten signaalien käsittelyssä sovelletaan erilaisia operaatioita ja algoritmeja. Käytännöllisesti katsoen useimmat perustuvat muutamisiin perusope-

raatioihin. Niitä ovat yhteenlasku (2.1), kertolasku (2.2), vakiolla kertominen (2.3) ja siirto ajassa (2.4):

$$y(n) = x_1(n) + x_2(n), \quad (2.1)$$

$$y(n) = x_1(n)x_2(n), \quad (2.2)$$

$$y(n) = ax(n) \text{ ja} \quad (2.3)$$

$$y(n) = x(n - d), \text{ missä } d \in \mathbb{Z}. \quad (2.4)$$

Esimerkiksi *konvoluutio* (convolution), joka on yksi tärkeimmistä DSP:n operaatioista, määritellään kaavalla

$$y(n) = x_1(n) * x_2(n) = \sum_{j=-\infty}^{\infty} x_1(j)x_2(n-j). \quad (2.5)$$

Oikealla puolella esiintyvän summalausekkeen muodossa operaatiosta käytetään myös nimitystä *konvoluutiosumma* (convolution sum). Se vastaa signaalin  $x_1(n)$  ja signaalin  $x_2(n)$  ajassa käännetyin version ristikorrelaatiota [24]. Konvoluutiota sovelletaan esimerkiksi digitaalisten suodattimien yhteydessä. Suodattimia käsitellään myöhemmin.

Kaavan (2.5) mukaisessa konvoluutiossa ei kiinnitetty huomiota signaalien pituuksiin. Käytännössä DSP:ssä signaalit ovat kuitenkin aina äärellisen pituisia. Kun signaalien  $x_1(n)$  ja  $x_2(n)$  pituuksia merkitään  $N_1$  ja  $N_2$ , voidaan konvoluutio määritellä kaavalla

$$y(n) = x_1(n) * x_2(n) = \sum_{j=0}^{N-1} x_1(j)x_2(n-j), \quad (2.6)$$

missä  $N = N_1 + N_2 - 1$  ja  $n$  saa arvot  $0, 1, \dots, N - 1$ . Signaaleista saadaan  $N$ :n pituisia lisäämällä sopiva määrä nollia molempien signaalien perään. Tämän asettelman mukaista operaatiota kutsutaan *lineaariseksi konvoluutioksi* (linear convolution). Jos ylimääräisiä nollia ei lisätä konvoloitaviin signaaleihin, on kyseessä *jaksollinen konvoluutio* (circular convolution). Käytännössä useimmissa tapauksissa ollaan kiinnostuneita juuri lineaarisesta konvoluutiosta. Smithin [40] mukaan jaksollista konvoluutiota pyritään välttämään.

## 2.2 Diskreettiaikaiset järjestelmät

Diskreettiaikainen järjestelmä on pohjimmiltaan matemaattinen algoritmi, joka vastaanottaa herätesignaalin  $x(n)$  ja tuottaa vastesignaalin  $y(n)$  [24]. Toteutukseltaan diskreettiaikainen järjestelmä voikin yhtä hyvin olla monimutkainen algoritmi kuin joukko yksinkertaisia laskutoimituksia. Teoreettisesti voidaan todeta,



että diskreettiaikainen järjestelmä määrittää kuvauksen herätesignaalien joukosta vastesignaalien joukkoon. Kun järjestelmän määrittämää kuvausta merkitään symbolilla  $\mathcal{S}$ , voidaan herätteen  $x(n)$  ja vasteen  $y(n)$  välinen suhde esittää yhtälöllä

$$y(n) = \mathcal{S}[x(n)].$$

Diskreettiaikaisia järjestelmiä luokitellaan Oppenheimin ja Schaferin [33] mukaan siten, että kuvaukselle  $\mathcal{S}$  asetetaan tiettyjä rajoituksia. Toisin sanoen luokittelu tehdään sen mukaan, mitä ominaisuuksia kuvauksen taustalla vaikuttavalla järjestelmällä on.

Olkoot  $x_1(n)$  ja  $x_2(n)$  toisistaan riippumattomia herätteitä,  $y_1(n)$  ja  $y_2(n)$  järjestelmän vasteita kyseisiin herätteisiin sekä  $a_1$  ja  $a_2$  mielivaltaisia vakioita. Diskreettiaikainen järjestelmä on *lineaarinen* (linear), jos ja vain jos

$$\mathcal{S}[a_1x_1(n) + a_2x_2(n)] = a_1\mathcal{S}[x_1(n)] + a_2\mathcal{S}[x_2(n)] = a_1y_1(n) + a_2y_2(n).$$

Lineaarisuudesta seuraa, että yhteenlasku ja vakiolla kertominen voidaan suorittaa ennen järjestelmän soveltamista tai sen jälkeen.

Diskreettiaikainen järjestelmä on *siirtainvariantti* (shift invariant) tai *aikainvariantti* (time invariant), jos herätteen siirtäminen  $d$ :llä aikayksiköllä aiheuttaa vastaavansuuruisen siirtymän järjestelmän vasteessa:

$$\text{jos } y(n) = \mathcal{S}[x(n)], \text{ niin } y(n-d) = \mathcal{S}[x(n-d)].$$

Ominaisuudesta seuraa, että siirto ajassa voidaan tehdä ennen siirtainvariantin järjestelmän soveltamista tai sen jälkeen.

Jos diskreettiaikainen järjestelmä on sekä lineaarinen että aikainvariantti, kutsutaan sitä *lineaariseksi aikainvariantiksi* (linear time invariant, LTI) järjestelmäksi. LTI-järjestelmän herätteen ja vasteen välinen suhde voidaan esittää konvoluutiosumman avulla:

$$y(n) = \mathcal{S}[x(n)] = \sum_{j=-\infty}^{\infty} h(j)x(n-j),$$

missä  $h(n)$  on järjestelmän *impulssivaste* (impulse response). LTI-järjestelmä voidaan täysin kuvata impulssivasteensa perusteella. Impulssivaste saadaan, kun herätteenä on *yksikköimpulssi* (unit impulse):

$$h(n) = \mathcal{S}[\delta(n)].$$

Yksikköimpulssi, jota kutsutaan myös *yksikkönäytteeksi* (unit sample), määritellään kaavalla

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}.$$

Diskreettiaikaista järjestelmää sanotaan *stabiiliksi* (stable), jos rajoitettu heräte aiheuttaa rajoitetun vasteen. Kun oletetaan, että herätearvot ovat rajoitettuja, on myös vaste rajoitettu, jos ja vain jos

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty.$$

Selvästi ehto on voimassa, jos impulssivaste on äärellisen pituinen. Muussa tapauksessa ehto on voimassa vain, jos impulssivasteen arvot lähestyvät nollaa  $n$ :n kasvaessa rajatta.

Diskreettiaikainen järjestelmä on *kausaallinen* (causal), jos järjestelmän vaste voidaan muodostaa yksinään nykyisen ja aikaisempien herätearvojen perusteella:

$$h(n) = 0, \text{ kun } n < 0.$$

Kausaalinen järjestelmä ei siis tarvitse tietoa tulevista herätearvoista vasteen muodostamiseksi. Tämä ominaisuus on tärkeä esimerkiksi reaaliaikaisissa järjestelmissä, joissa herätearvot ovat luonnollisesti saatavilla vasta niiden mittaamisen jälkeen. On syytä huomioida, että ei-kausaalisen järjestelmän toteuttaminen reaaliaikaisena on useimmiten mahdotonta [12].

LTI-järjestelmät jaetaan impulssivasteen pituuden perusteella kahteen luokkaan: *äärellisen impulssivasteen* (finite impulse response, FIR) ja *äärettömän impulssivasteen* (infinite impulse response, IIR) järjestelmiin. Teoriassa molempien järjestelmätyyppien toiminta voidaan mallintaa impulssivasteen ja konvoluutiosumman avulla. Käytännössä tämä onnistuu ainoastaan FIR-järjestelmien tapauksessa, sillä IIR-järjestelmän impulssivaste on äärettömän pituinen. Tavallisesti IIR-järjestelmä esitetäänkin rekursiivisessa muodossa kaavalla

$$y(n) = \sum_{j=0}^{N-1} b_j x(n-j) - \sum_{k=1}^{M-1} a_k y(n-k), \quad (2.7)$$

missä  $b_j$  ja  $a_k$  ovat järjestelmän kertoimia. Kun kaavan (2.7) mielivaltainen järjestelmä oletetaan kausaaliseksi, niin huomataan IIR-järjestelmän vasteen muodostuvan nykyisen herätearvon sekä edellisten heräte- ja vastearvojen perusteella. Kausaalisen FIR-järjestelmän vaste muodostuu yksinään nykyisen ja edellisten herätearvojen perusteella. Jos siis kaavan (2.7) kertoimet  $a_k$  asetetaan nolliksi, on kyseessä FIR-järjestelmä:

$$y(n) = \sum_{j=0}^{N-1} b_j x(n-j).$$

Nyt kertoimet  $b_j$  muodostavat FIR-järjestelmän impulssivasteen, jonka pituus on  $N$  näytettä.

*Suodatin* (filter) on järjestelmä, joka muuttaa signaalin ominaisuuksia jollain tietyllä tavalla. Usein suodatuksen tavoitteena on signaalin laadun parantaminen, mutta suodattimia voidaan suunnitella hyvin erilaisiin tehtäviin. Sen tähden termit järjestelmä ja suodatin ovatkin kirjallisuudessa useimmiten vaihtokelpoisia. Tässä kohdassa ei perehdytä syvällisesti suodattimiin tai niiden suunnitteluun, koska aiheita käsitellään kattavasti kirjallisuudessa. Diniz *et al.* [12] käsittelevät pääasiassa digitaalisten järjestelmien suunnittelua ja analysointia. Hamming [20] ja Taylor [45] keskittyvät erityisesti digitaalisiin suodattimiin. Edellisten lisäksi aiheita käsitellään lähteissä [24], [33] ja [40].

Lineaaristen suodattimien toimintaa voidaan tutkia *taajuusvasteen* (frequency response) avulla. Taajuusvaste kuvaa, kuinka suodatin muuttaa eri taajuuksisten sinimuotoisten signaalien amplitudia ja vaihetta. Suodattimet suunnitellaan usein siten, että asetetaan vaatimuksia niiden toiminnalle signaalin taajuusformaation suhteen. Jos esimerkiksi tiedetään häiriön esiintyvän tietyllä taajuuskaistalla, voidaan suunnitella suodatin, joka vaimentaa signaalin taajuuskomponentteja kyseisellä kaistalla.

Taajuusvalintaiset suodattimet jaetaan usein neljään luokkaan. *Alipäästösuodatin* (lowpass filter) vaimentaa määritellyä taajuutta korkeampia signaalin taajuuskomponentteja. *Ylipäästösuodatin* (highpass filter) toimii päinvastoin, eli se vaimentaa määritellyä taajuutta matalampia taajuuskomponentteja. *Kaistanestosuodatin* (bandstop filter) vaimentaa tietyn taajuuskaistan alueella olevat taajuuskomponentit, ja *kaistanpäästösuodatin* (bandpass filter) vaimentaa määritellyn taajuuskaistan ulkopuolisia taajuuskomponentteja. Toisinaan puhutaan myös *mielivaltaisen taajuusvasteen* (arbitrary frequency response) suodattimesta, jos sitä ei voida taajuusvasteensa perusteella luonnehtia yhdeksikään edellisen tyyppisistä suodattimista.

Käytännössä suodattimen suunnittelu saattaa olla vaikeampaa kuin edellisen perusteella voisi ymmärtää. Signaalin siirtämä tärkeä informaatio voi nimittäin sijaita osittain tai kokonaan samalla taajuuskaistalla kuin häiriökin. Tämä vaikeuttaa huomattavasti sopivan suodattimen suunnittelua. Ratkaisu ei välttämättä löydykään perinteisistä lineaarisista suodattimista, vaan toisinaan joudutaan soveltamaan epälineaarisia signaalinkäsittelyn menetelmiä.

## 2.3 Diskreetti Fourier-muunnos

Luvun alussa diskreetti-aikainen signaali määriteltiin kaksisuuntaisena lukujonona, jonka termit vastaavat signaalin näytearvoja. Kun signaaleja tarkastellaan ajan funktiona muuttuvina arvoina, sanotaan niiden olevan *aikatasossa* (time domain). Aikatasoesityksen lisäksi signaaleja voidaan tarkastella myös vaihtoehtoisilla tavoilla, kuten taajuuden funktiona. Kun signaali esitetään taajuuden funktiona,

sanotaan sen olevan *taajuustasossa* (frequency domain).

Käsiteltävät signaalit ovat useimmiten aikatasossa. Siksi tarvitaan menetelmä, jonka avulla voidaan muodostaa signaalin taajuusesitys. Kyseiseen tehtävään voidaan käyttää *Fourier-muunnosta* (Fourier transform). Siitä on neljä versiota, joista jokainen on määritelty eri tyyppisille signaaleille (katso esim. [40]). Koska DSP:ssä voidaan käyttää pääasiassa vain *diskreettiä Fourier-muunnosta* (discrete Fourier transform, DFT), tarkastellaan jatkossa pelkästään sitä. DFT:ssä signaali esitetään sopivasti painotettujen ja ajassa siirrettyjen kompleksisten eksponenttifunktioiden summana.

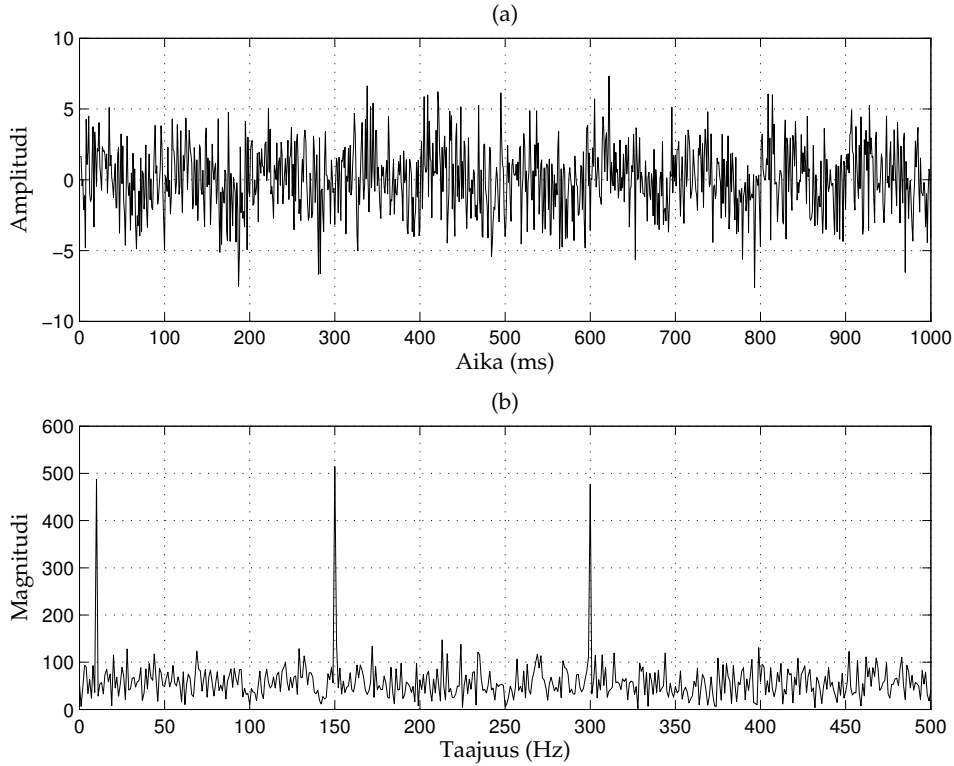
DFT on tarkalleen ottaen määritelty vain jaksollisille signaaleille. Diskreettiai-kainen signaali on *jaksollinen* (periodic), jos on olemassa sellainen kokonaisluku  $N$ , että kaikilla kokonaisluvulla  $n$  on voimassa  $x(n) = x(n + N)$ . Tällöin lukua  $N$  kutsutaan signaalin jaksoksi tai jakson pituudeksi. Jaksollisen signaalin  $x(n)$  diskreetti Fourier-muunnos määritellään kaavalla

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-i\frac{2\pi}{N}kn}, \quad (2.8)$$

missä  $N$  on jakson pituus,  $i$  on imaginaariyksikkö ja  $k$  saa arvot  $0, 1, 2, \dots, N-1$ . DFT:tä voidaan soveltaa myös äärellisen pituisten signaalien yhteydessä, vaikka ne eivät olekaan jaksollisia. DFT:n kannalta äärellisen pituinen signaali voidaan nimittäin käsittää jaksollisen signaalin yksittäisenä jaksona. Diniz *et al.* [12] mukaan DFT voidaankin käsittää kahdella tavalla: joko  $N$ :n pituisen tai  $N$ -jaksollisen signaalin taajuusesityksenä.

Diskreetin Fourier-muunnoksen tuloksena on kompleksinen lukujono, mikä kuvaa alkuperäisen signaalin taajuustasossa. Usein halutaan tarkastella DFT:n itseisarvoa ja vaihetta. Ne määritellään harmonisille komponenteille  $X(k)$  samalla tavalla kuin itseisarvo ja vaihekulma kompleksiluvuille. Koska DFT on symmetrinen kohdan  $N/2$  suhteen, riittää kun signaalin taajuusinformaatiota analysoidessa tarkastellaan  $k$ :n arvoja  $0, 1, \dots, N/2$ . Harmoninen komponentti  $X(N/2)$  vastaa korkeinta signaalissa esiintyvää taajuuskomponenttia, joka on puolet näytteenottotaajuudesta. Kun näytteenottotaajuus on  $F_s$  Hz, niin yleisesti harmonista komponenttia  $X(k)$  vastaava taajuus on  $k(F_s/N)$  Hz.

Ifeachor ja Jervis [24] mainitsevat Fourier-muunnoksen eräänä hyötynä sen, että muunnoksen seurauksena tulkintojen tekeminen saattaa helpottua. Esimerkiksi kuvan 2.1 (a) signaali koostuu useista sinimuotoisista aaltofunktioista ja normaalisti jakautuneesta kohinasta. Kuvassa 2.1 (b) on signaalin Fourier-muunnoksen itseisarvo, mistä voidaan selvästi havaita signaalissa esiintyvät taajuuskomponentit. Tässä tapauksessa hallitsevat komponentit esiintyvät 10, 150 ja 300 Hz:n taajuuksilla (näytteenottotaajuus on 1 kHz). Kyseessä on hyvin yksinkertainen esimerkki, mutta se osoittaa, että toisinaan voi olla merkityksellisempää tarkastella signaalia taajuustasossa.



Kuva 2.1: Yhden sekunnin mittainen signaali (a) ja sen Fourier-muunnoksen itseisarvo (b). Fourier-muunnoksen itseisarvon perusteella signaalin hallitsevat taajuuskomponentit esiintyvät 10, 150 ja 300 Hz:n taajuuksilla.

Fourier-muunnos on kääntyvä operaatio, jolloin taajuustasossa oleva signaali voidaan muuntaa aikatasoon. Erityisesti *käänteinen diskreetti Fourier-muunnos* (inverse discrete Fourier transform, IDFT) määritellään kaavalla

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{i \frac{2\pi}{N} kn}, \quad n = 0, 1, \dots, N-1.$$

DFT ja IDFT eivät sovellu kovinkaan hyvin reaaliaikaiseen signaalinkäsittelyyn, koska pitkillä signaaleilla niiden aikakompleksisuus kasvaa neliöllisesti signaalin pituuden suhteen. Esimerkiksi DFT:n tapauksessa tämä voidaan todeta tarkastelemalla kaavaa (2.8), missä jokaista harmonista komponenttia  $X(k)$  kohden  $n$  käy läpi arvot  $0, 1, 2, \dots, N-1$ . Kun harmonisia komponentteja on vastaava määrä, saadaan DFT:n aikakompleksisuudeksi  $O(N^2)$ , missä  $N$  on siis signaalin jakson pituus. Vuonna 1965 Cooley ja Tukey julkaisivat algoritmin, jolla DFT voidaan laskea huomattavasti nopeammin. Algoritmin aikakompleksisuus on

$O(N \log_2 N)$ . Yleisesti algoritmia jolla DFT voidaan laskea nopeasti kutsutaan *nopeaksi Fourier-muunnokseksi* (fast Fourier transform, FFT).

Taajuusanalyysin ohella eräs FFT:n sovellus on konvoluution nopea laskenta. *Konvoluutioteoreeman* (convolution theorem) mukaan konvoluutio aikatasossa vastaa kertolaskua taajuustasossa. Kun Fourier-muunnosta merkitään symbolilla  $\mathcal{F}$  ja käänteistä Fourier-muunnosta symbolilla  $\mathcal{F}^{-1}$ , voidaan konvoluutioteoreeman perusteella kirjoittaa yhtälö

$$x_1(n) * x_2(n) = \mathcal{F}^{-1}[\mathcal{F}[x_1(n)]\mathcal{F}[x_2(n)]].$$

Nopeushyöty konvoluution suoraan laskentaan verrattuna esiintyy tosin vain silloin, kun konvoloitavat signaalit ovat riittävän pitkiä. Fourier-muunnoksen teoriaa, FFT-algoritmeja ja nopean konvoluution toteuttamista käsitellään yksityiskohtaisemmin kirjallisuudessa (katso esim. [12], [24], [33] ja [40]). Cormen *et al.* [7] käsittelevät FFT:tä algoritmitutkimuksen näkökulmasta.

## 2.4 Aaloke-muunnos

Diskreetissä Fourier-muunnoksessa kantafunktioina käytetään kompleksisia eksponenttifunktioita, jotka vaikuttavat kaikilla ajanhetkillä. Sen tähden DFT soveltuu huonosti signaalin hetkellisten piirteiden tarkasteluun. Sitä voidaan soveltaa, jos ollaan kiinnostuneita vain signaalin yleisestä luonteesta. Toisinaan tarvitaan kuitenkin kuvaus, jonka avulla on mahdollista erottaa signaalin taajuusinformaation muutokset ajan kuluessa. Tämän kaltainen tarve voi esiintyä esimerkiksi silloin, kun tarkastellaan *epästationaarisia* (non-stationary) signaaleja, joiden ominaisuudet muuttuvat ajan funktiona. Signaali voidaankin esittää sekä ajan että taajuuden funktiona, milloin signaalin sanotaan olevan *aika-taajuustasossa* (time-frequency domain).

Aika-taajuusesityksen muodostamiseen voidaan käyttää esimerkiksi *lyhyen aikavälin Fourier-muunnosta* (short-time Fourier transform, STFT), jota kutsutaan myös *ikkunoiduksi Fourier-muunnokseksi* (windowed Fourier transform). Diskreettiä aikaisen jaksollisen signaalin  $x(n)$  lyhyen aikavälin Fourier-muunnos määritellään kaavalla

$$X(t, k) = \sum_{n=0}^{N-1} x(n)w(n-t)e^{-i\frac{2\pi}{N}kn},$$

missä  $t = 0, 1, \dots, N-1$  ja jokaista  $t$ :tä kohden  $k$  saa arvot  $0, 1, \dots, N-1$ . Kaavassa esiintyy ikkunafunktio  $w(n)$ , jonka avulla muunnos kohdistetaan signaalin tietylle ajanjaksolle. Komponentti  $X(t, k)$  voidaankin tulkita harmonisena komponenttina

$X(k)$ , joka vaikuttaa tietyllä ajanjaksolla. Käänteinen STFT määritellään kaavalla

$$x(n) = \frac{1}{N} \sum_{t=0}^{N-1} w(n-t) \sum_{k=0}^{N-1} X(t,k) e^{i\frac{2\pi}{N}kn},$$

missä  $n$  saa arvot  $0, 1, \dots, N-1$ . Käytetty ikkunafunktio vaikuttaa olennaisesti muunnoksen aika- ja taajuusresoluutioon ja näin myös muunnoksen tulokseen. Muiden muassa Mallat [30] ja Polikar [36] käsittelevät sopivan ikkunafunktion valitsemista.

*Heisenbergin epätarkkuusperiaate* (Heisenberg Uncertainty Principle) sanoo, ettei vapaan hiukkasen sijaintia ja liikemäärää voida määrittää yhtä aikaa mielivaltaisen tarkasti. Vaikka epätarkkuusperiaate onkin lähtöisin hiukkasfysiikasta, vaikuttaa se myös muilla tieteenaloilla signaalinkäsittely mukaan lukien. Erityisesti se rajoittaa aika- ja taajuusresoluution tarkkuutta STFT:ssä; mitä tarkempi aikaresoluutio on, sitä epätarkempi on taajuusresoluutio, ja päinvastoin. Epätarkkuusperiaatteesta seuraa, että ikkunafunktion äärellinen pituus rajoittaa taajuusresoluutiota. Toisaalta ikkunafunktion pituuden kasvaessa aikaresoluutio huononee. Polikar [36] toteaaakin, että äärettömän pituista ikkunaa käytettäessä STFT palautuu tavalliseksi Fourier-muunnokseksi. Tällöin menetetään kokonaan muunnoksen aikainformaatio.

*Aaloke-muunnoksessa* (wavelet transform) epätarkkuusperiaatteesta seuraavat ongelmat pyritään ratkaisemaan siten, että aika- ja taajuusresoluutiota muutetaan muunnoksen aikana. Aaloke-muunnoksessa ei sovelleta kompleksisia eksponenttifunktioita, kuten Fourier-menetelmissä, vaan *aallokkeiksi* (wavelet) kutsuttuja aaltofunktioita. Ne ovat lokalisoituja ajassa, päinvastoin kuin kompleksiset eksponenttifunktiot. Aaloke-muunnoksen kantafunktiot muodostetaan niin sanotun *äitiaallokkeen* (mother wavelet) skaalauksina ja translaatioina. Translaatiolla tarkoitetaan aallokkeen siirtämistä ajassa, mikä vastaa STFT:ssä muunnoksen kohdistamista tietylle ajanjaksolle. Skaalauksella aalloketta voidaan levittää siten, että muunnoksen aika- ja taajuusresoluutio muuttuvat. Muunnoksessa voidaan käyttää erilaisia äitiaallokkeita (katso esim. [30]).

Aaloke-muunnoksessa korkeita taajuuksia analysoidaan paremmalla aikaresoluutiolla ja matalia taajuuksia vastaavasti paremmalla taajuusresoluutiolla. Diskreettiaikaisen signaalin aaloke-muunnos lasketaan tavallisesti pyramidalgoritmilta. Siinä sovelletaan äitiaallokkeeseen perustuvia suodattimia, sekä näytteenotto-taajuuden muuttamista ali- ja ylinäytteistyksellä. Aaloke-muunnosta ja pyramidalgoritmia tarkastellaan lähteissä [12], [19], [24] ja [36]. Mallat [30] käsittelee yksityiskohtaisesti myös aallokkeiden teoriaa. Seuraavaksi esitellään pyramidalgoritmin toiminta pääpiirteissään.

Diskreettiaikaisen signaalin aaloke-muunnoksessa käytetään äitiaallokkeeseen perustuvia suodattimia  $H$  ja  $G$ . Ensimmäinen on alipäästösuodatin ja jälkimmäi-

nen on ylipäästösuodatin. Syötesignaali  $x(n)$ , jonka pituus on  $N$  näytettä, suodatetaan alipäästösuodattimella  $H$ . Suodatettu signaali alinäytteistetään kahdella (joka toinen näytearvo poistetaan), minkä seurauksena näytteenottotaajuus puolittuu. Tuloksena on  $N/2$ :n näytteen pituinen tulossignaali  $a_1(n)$ . Se käsittää syötesignaalin yleispiirteitä ja sitä kutsutaankin syötesignaalin *approksimaatioksi* (approximation). Syötesignaali suodatetaan myös ylipäästösuodattimella  $G$  ja alinäytteistetään kuten edellä. Tuloksena on  $N/2$ :n näytteen pituinen tulossignaali  $d_1(n)$ , joka sisältää *aallope-kertoimet* (wavelet coefficients).

Edellä kuvattua vaihetta toistetaan rekursiivisesti edellisen tason approksimaatiolle  $a_l(n)$ . Tasot  $l$  vastaavat skaaloja  $2^l$ . Laskenta pysähtyy, kun saavutetaan ennalta määrätty taso tai kun syötesignaalia ei voi enää alinäytteistää. Pyramidialgoritmin tuloksena on siis syötesignaalin esitys eri tasojen aallope-kertoimina ja lisäksi viimeisen tason approksimaationa. Pyramidialgoritmin toimintaa aallope-muunnoksessa on havainnollistettu kuvassa 2.2.

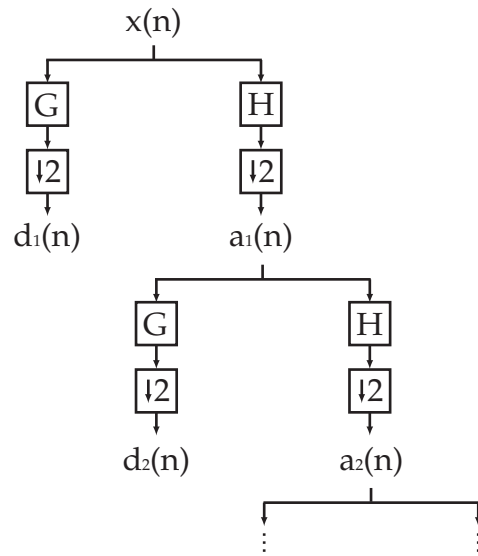
Käänteisessä aallope-muunnoksessa käytetään vastaavasti rekonstruktiosuodattimia  $H_r$  ja  $G_r$ . Tason  $l$  approksimaatio  $a_l(n)$ , jonka pituus on  $N$  näytettä, ylinäytteistetään kahdella (signaalin jokaisen näytearvon perään lisätään yksi uusi näytearvo, joka on arvoltaan nolla). Tuloksena on  $2N$ :n näytteen pituinen signaali, jonka näytteenottotaajuus on kaksinkertainen alkuperäiseen signaaliin verrattuna. Ylinäytteistetty signaali suodatetaan rekonstruktiosuodattimella  $H_r$ . Myös saman tason aallope-kertoimista koostuva signaali  $d_l(n)$  ylinäytteistetään kuten edellä ja suodatetaan rekonstruktiosuodattimella  $G_r$ . Tulossignaalit lasketaan yhteen näytteittäin, jolloin saadaan tason  $l - 1$  approksimaatio  $a_{l-1}(n)$ .

Edellä kuvattua vaihetta toistetaan rekursiivisesti kunnes voidaan muodostaa tason 0 approksimaatio. Se vastaa alkuperäistä signaalia. Pyramidialgoritmin toiminta käänteisessä aallope-muunnoksessa esitetään kuvassa 2.3.

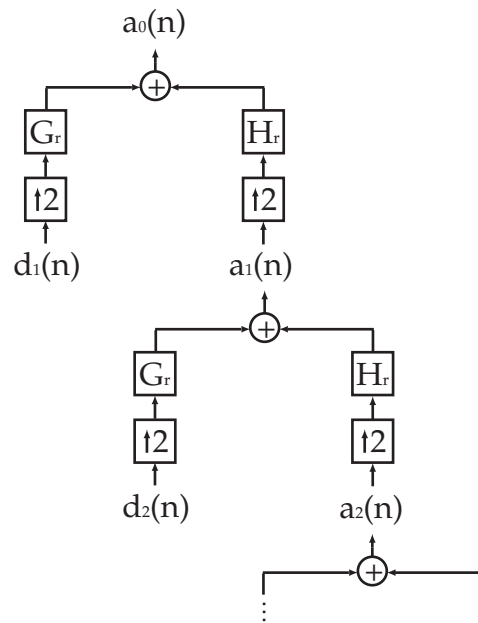
Signaalien analysoimisen lisäksi aallope-muunnosta voidaan käyttää myös toisenlaisissa sovelluksissa. Eräs sovelluskohde on tiedon tiivistäminen. Aallokkeisiin perustuvaa kuvien pakkaamista on käytetty esimerkiksi FBI:n sormenjälkitietokannoissa [19]. Aallope-muunnosta voidaan soveltaa myös signaalissa esiintyvän kohinan vaimentamisessa. Tässä tutkielmassa *aallope-kohinanpoistoa* (wavelet denoising) kokeiltiin vaihtoehtoisena esikäsittelymenetelmänä perinteisen kaistanpäästösuodatuksen ohella. Mallat [30] käsittelee aallope-muunnoksen soveltamista kohinanpoistossa ja tiedon tiivistämisessä.

Aallope-kohinanpoiston toimintaperiaate on seuraavanlainen. Ensin käsiteltävä signaali muunnetaan pyramidialgoritmillä ennalta määrätylle tasolle asti. Sen jälkeen jokaisen tason aallope-kertoimet kynnystetään sopivasti valittujen tasokohdistusten kynnysten mukaan. Tässä kohdassa tuleekin soveltaa sopivia menetelmiä kynnysarvojen laskemiseen sekä aallope-kertoimien kynnystämiseen. Käsillä olevan työn kannalta keskeisessä asemassa ovat erityisesti SURE (Stein Unbiased Risk Es-





Kuva 2.2: Pyramidialgoritmin toiminta aaloke-muunnoksessa.



Kuva 2.3: Pyramidialgoritmin toiminta käänteisessä aaloke-muunnoksessa.

timator) -kynnykset ja *pehmeä kynnystys* (soft thresholding), koska juuri niitä sovellettiin tässä tutkielmassa. Lopuksi kynnystetyt aaloke-kertoimet ja viimeisen tason approksimaatio muunnetaan takaisin aikatasoon pyramidalgoritmin käänteisellä versiolla. Seuraavaksi käsitellään edellä mainittuja menetelmiä lähteen [30] esitystä mukaillen.

SURE-kynnykset määritellään tasokohtaisesti aaloke-kertoimien perusteella. Tarkastellaan aaloke-muunnoksen tasoa  $l$ . Ensin aaloke-kertoimien  $d_l(n)$  perusteella muodostetaan lukujono  $D_l(k)$ , missä aaloke-kertoimet ovat itseisarvonsa perusteella laskevassa suuruusjärjestyksessä. Muuttuja  $k$  saa arvot  $1, 2, \dots, N$ , kun tasolla  $l$  on  $N$  kappaletta aaloke-kertoimia. Nyt  $D_l(1)$  on itseisarvoltaan suurin ja  $D_l(N)$  vastaavasti itseisarvoltaan pienin aaloke-kerroin. Seuraavaksi lasketaan riskiestimaattorin

$$R(k) = \sum_{j=k}^N |D_l(j)|^2 - (N - k)\sigma^2 + k(\sigma^2 + |D_l(k)|) \quad (2.9)$$

arvot muuttujan  $k$  arvoilla  $1, 2, \dots, N$ . Kynnykseksi valitaan sellainen  $|D_l(k)|$ , jota vastaavalla  $k$ :n arvolla riskiestimaattori saa minimiarvonsa. Useimmiten tehdään oletus, jonka mukaan signaalissa vaikuttava kohina on peräisin normaalijakaumasta. Silloin varianssi ( $\sigma^2$ ) saa kaavassa (2.9) arvon yksi.

Kun tasokohtainen kynnys on määritetty, voidaan aaloke-kertoimet kynnystää sen perusteella. Merkitään kyseessä olevan tason kynnystettävää aaloke-kerrointa symbolilla  $d$ . Nyt kynnystetty aaloke-kerroin  $d_t$  määritellään pehmeän kynnystyksen mukaan kaavalla

$$d_t = \begin{cases} d - T, & d > T \\ d + T, & d < -T \\ 0, & |d| \leq T \end{cases},$$

missä  $T$  on tasokohtainen kynnysarvo. Se voidaan määrittää esimerkiksi edellä kuvatulla tavalla. Silloin kyseessä on SURE-kynnys.

## Luku 3

# Neuroverkot

*Neuroverkot* (neural network) ovat hajautettuja rinnakkaisprosessoivia järjestelmiä, jotka kykenevät oppimaan toimintansa käsiteltävän informaation perusteella. Keinotekkoisten neuroverkkojen esikuvina ovat toimineet biologiset neuroverkot ja erityisesti ihmisaivot. Haykinin [21] mukaan neuroverkkotutkimusta onkin motivoinut alusta alkaen se, että ihmisaivot käsittelevät tietoa eri tavalla kuin tavanomaiset tietokoneet. Hajautetun rakenteensa ja rinnakkaisprosessoivan toimintansa johdosta aivot voivat suorittaa tiettyjä tehtäviä tarkemmin ja nopeammin kuin nykyaikaiset tietokoneet. Esimerkiksi puheentunnistus saattaa tuntua ihmisestä kovin arkipäiväiseltä, mutta tietokoneelle se on vielä nykyään huomattavasti vaikeampi tehtävä. Vaikka nykyaikaiset tietokoneet ovatkin tehokkaita työjuhtia vaativien laskutoimitusten laskennassa, ei se yksinään riitä puheentunnistuksen kaltaisten ongelmien ratkaisemiseen.

Ihmisten kyky ymmärtää puhetta, tai tuottaa sitä, ei ole kuitenkaan synnynäinen ominaisuus. Se on kehittynyt kokemukseen perustuvan oppimisprosessin tuloksena. Aivojen tavoin myös neuroverkoilla on kyky oppia ja mukautua toimimaan erilaisissa tilanteissa. Oppimiskyvyn lisäksi eräs toinen aivojen ja neuroverkkojen yhteinen ominaisuus on yleistämiskyky. Sen vuoksi ihminen voi reagoida uusiin tilanteisiin aikaisemmin opitun tiedon perusteella. Yleistämiskyvyn johdosta myös neuroverkot voivat reagoida mielekkäästi sellaisiin syötteisiin, joita ne eivät ole kohdanneet oppimisen aikana.

Neuroverkoilla pyritään mallintamaan abstraktilla tasolla joitain aivojen toimintoja. On kuitenkin syytä muistaa neuroverkkojen olevan vain karkeita malleja biologisista vastineistaan, vaikka ne aivoja joltain osin muistuttaisivatkin. Lisäksi Fu [17] huomauttaa, että keinotekoisissa neuroverkoissa sovelletut laskennalliset menetelmät eivät aina ole edes selitettävissä, kun niitä lähestytään biologisista näkökulmista. Neuroverkkoja tutkivaa tieteenalaa kutsutaan *neurolaskennaksi* (neurocomputing). Tässä luvussa käsitellään joitain sen keskeisimpiä aiheita. Tarkemmin perehdytään monikerrosperceptron-neuroverkkoihin, koska niitä sovellettiin

tässä tutkielmassa.

## 3.1 Neuronit

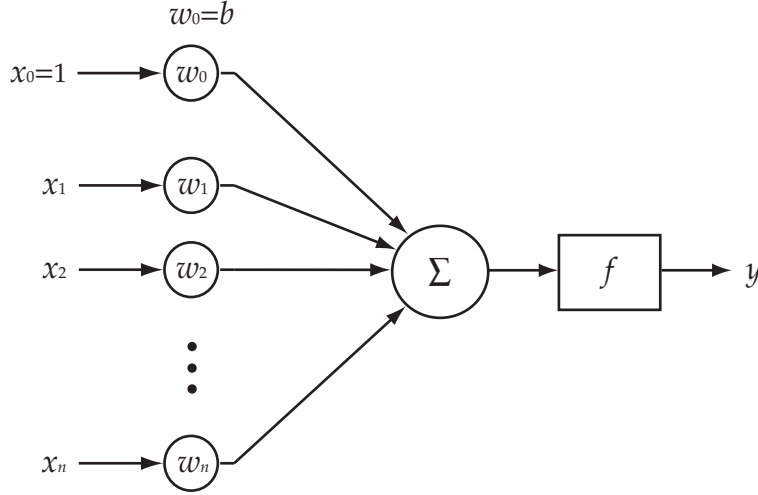
*Hermosolut* eli *neuronit* (neuron) ovat aivojen perusyksiköitä. Ne koostuvat solukeskuksesta, *tuojahaarakkeista* eli *dendriiteistä* (dendrite) ja *viejähaarakkeesta* eli *aksonista* (axon). Dendriitit vastaanottavat hermosoluun saapuvia hermoimpulsseja ja aksoni välittää lähteviä hermoimpulsseja muille neuroneille. *Synapsit* (synapse) ovat kahden hermosolun välisiä liitoskohtia. Tyypillisimpiä ovat kemialliset synapsit, joissa hermoimpulssien välittyminen perustuu sähkökemialliseen vuorovaikutukseen [21]. Hermosolu aktivoituu tai laukeaa, kun sen vastaanottama kokonaisheräte on riittävän suuri. Aktivaatiosta syntynyt hermoimpulssi etenee aksonia pitkin eteenpäin.

Neuroverkkojen perusyksiköitä ovat vastaavasti keinotekoiset neuronit. Niiden rakenne muistuttaa karkeasti biologisten neuronien rakennetta. Myös keinotekoisien neuronin toimintaperiaate mukailee biologisen neuronin toimintaa. Syöte välittyy neuronille syötekanavien eli synapsien kautta. Niistä jokaiseen liittyy *synaptinen paino* (synaptic weight), joka määrittää voimakkuuden, jolla synapsin herätearvo vaikuttaa neuronin toimintaan. *Summain* (adder) laskee synaptisilla painoilla painotettujen herätearvojen summan, joka välittyy edelleen neuronin *aktivaatiofunktio*lle (activation function). Se rajoittaa summaimen laskeman tuloksen äärelliselle lukuvälille ja muodostaa neuronin varsinaisen tuloksen. Neuronin liittyy tavallisesti myös *siirto-termi* (bias), jota vaihtelemalla voidaan vaikuttaa aktivaatiofunktioille välittyvän kokonaisherätteen suuruuteen. Toisinaan siirtotermiä kutsutaan myös kynnykseksi tai kynnysarvoksi.

Neuronin toiminta on käytännöllisempää mallintaa matemaattisesti. Merkitään neuronin synapseihin saapuvia herätearvoja  $x_1, x_2, \dots, x_n$  sekä neuronin synaptisia painoja  $w_1, w_2, \dots, w_n$ . Summain laskee neuronin herätearvojen painotetun summan. Jokaisen synapsin herätearvo  $x_i$  kerrotaan vastaavalla painolla  $w_i$  ja näin saadut arvot lasketaan yhteen. Neuronin kokonaisherätteen suuruuteen vaikuttaa myös neuronin liittyvä siirtotermi  $b$ . Herätearvojen painotetun summan ja siirtotermin muodostama kokonaisheräte välittyy aktivaatiofunktioille  $f$ , joka antaa neuronin varsinaisen tuloksen:

$$y = f \left( \sum_{i=1}^n w_i x_i + b \right). \quad (3.1)$$

Siirtotermin vaikutus voidaan huomioida myös kaavan (3.1) summalausekkeen laskennassa. Määritellään neuronin vakiosyöte  $x_0 = 1$  ja sitä vastaava painokerroin



Kuva 3.1: Kaavan (3.2) mukaisen neuronin toiminta.

$w_0 = b$ . Nyt neuronin syötteen ja tuloksen välinen suhde voidaan esittää kaavalla

$$y = f \left( \sum_{i=0}^n w_i x_i \right). \quad (3.2)$$

Kaavan (3.2) mukaisen neuronin toimintaa on havainnollistettu kuvassa 3.1. Edellä esiintyneen aktivaatiofunktion  $f$  tilalla voidaan käyttää eri tyyppisiä funktioita. Niiden tarkoituksena on rajoittaa neuronin tulos äärelliselle lukuvälille (tavallisesti  $[0, 1]$  tai  $[-1, 1]$ ).

Yksinkertaisin aktivaatiofunktio on *kynnysfunktio* (threshold function) tai *Heavisiden askelfunktio* (Heaviside step function). Kun aktivaatiofunktioille välitettävää kokonaisherätettä merkitään muuttujalla  $s$ , voidaan kynnysfunktio määritellä kaavalla

$$f(s) = \begin{cases} 1, & s \geq 0 \\ 0, & s < 0 \end{cases}.$$

Tällöin neuronin tulos on 1 kun kokonaisheräte on ei-negatiivinen ja muulloin tulos on 0. Toisinaan tuloksen on rajoituttava lukuvälille  $[-1, 1]$ . Silloin kynnysfunktio voidaan määritellä kaavalla

$$f(s) = \begin{cases} +1, & s \geq 0 \\ -1, & s < 0 \end{cases}.$$

Neuronia jossa käytetään aktivaatiofunktiona kynnysfunktioita, kutsutaan kirjallisuudessa McCulloch-Pitts -tyyppiseksi (katso esim. [3], [21] ja [29]). Kynnysfunktion luonteen perusteella McCulloch-Pitts -neuroninoudattaa 'kaikki tai ei mitään'

-periaatetta. Sen mukaan neuronin ei anna positiivista tulosta, jos sen vastaanottama syöte ei ole riittävän suuri. Unipolaarisen ja bipolaarisen kynnysfunktion kuvaajat esitetään kuvissa 3.2 (a) ja 3.2 (b).

*Sigmoidiset funktiot* (sigmoid function) ovat neuroverkkojen yhteydessä yleisimmin käytettyjä aktivaatiofunktioita [21]. Ne ovat sekä jatkuvia että derivoituvia koko reaalilukualueella (päinvastoin kuin kynnysfunktio). *Logistinen funktio* (logistic function), mikä saa arvoja lukuväliltä  $(0, 1)$ , on eräs sigmoidinen aktivaatiofunktio. Se määritellään kaavalla

$$f(s) = \frac{1}{1 + e^{-as}},$$

missä parametri  $a$  vaikuttaa funktion kaltevuuteen. Bishopin [3] mukaan logistisen aktivaatiofunktion käyttämistä motivoi erityisesti se, että sen tuottamat tulokset voidaan tulkita todennäköisyyksinä. Toisinaan tarvitaan sigmoidinen aktivaatiofunktio, jonka arvot sijoittuvat lukuvälille  $(-1, 1)$ . Tällöin voidaan käyttää *hyperbolista tangenttifunktiota* (hyperbolic tangent function), joka määritellään kaavalla

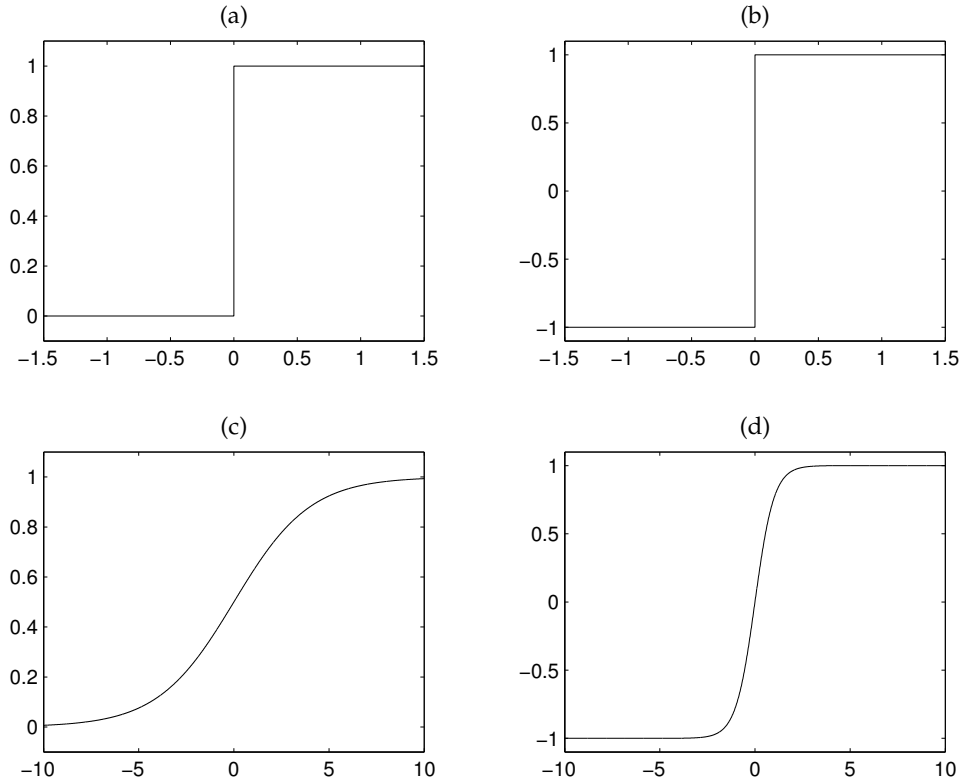
$$f(s) = \tanh(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}.$$

Sekä logistisen funktion että hyperbolisen tangenttifunktion kuvaajat esitetään kuvissa 3.2 (c) ja 3.2 (d).

## 3.2 Neuroverkkoarkkitehtuureja

Keinotekoiset neuroverkot rakentuvat ihmisaivojen tavoin toisiinsa yhdistetyistä neuroneista. Neuroneja yhdistämällä onkin mahdollista muodostaa räätälöityjä neuroverkkoja erilaisiin tarpeisiin. Kun neuroneissa sovelletaan eri tyyppisiä aktivaatiofunktioita, voidaan käytännöllisesti katsoen muodostaa lähes mielivaltaisia neuroverkkoja. Siitä huolimatta monimutkaisimpien aivojen rakenteiden mallintaminen neuroverkoilla on vielä tällä hetkellä vaikeaa [21].

Tavallisesti neuroverkot mielletään kerroksittaisiksi rakenteiksi, joissa neuronit on järjestetty tehtäviensä perusteella eri kerroksiin. Neuroverkon syötteitä vastaanottavat neuronit kuuluvat *syötekerrokseen* (input layer). Syötekerroksen neuroneihin ei liity painokertoimia, summainta, aktivaatiofunktioita eikä siirtotermiä. Syötteen vastaanottamisen lisäksi niiden ainoana tehtävänä on syötteen välittäminen seuraavien kerrosten neuroneille. Verkon laskennalliset neuronit, jotka toimivat kohdassa 3.1 esitetyllä tavalla, kuuluvat *tuloskerrokseen* (output layer) sekä mahdollisesti yhteen tai useampaan *piilokerrokseen* (hidden layer). Piilokerrosten neuroneista ei ole välittömiä yhteyksiä verkon ulkopuolelle. Tuloskerrokseen kuuluvat sellaiset neuronit, jotka muodostavat neuroverkon tuloksen. Neuronien väliset



Kuva 3.2: Aktivaatiofunktioita: unipolaarinen kynnysfunktio (a), bipolaarinen kynnysfunktio (b), logistinen funktio (c) missä  $a = 0.5$  ja hyperbolinen tangentti-funktio (d).

yhteydet voidaan määritellä mielivaltaisesti, ja ne saattavatkin muodostaa niin monimutkaisia rakenteita, että toisinaan edellä kuvattua neuroverkon kerroksittaista rakennetta on vaikea hahmottaa.

Neuroverkkoja voidaan luokitella niiden rakenteellisesta monimuotoisuudesta huolimatta. Yksi tapa lähestyä neuroverkkojen luokittelua, on tarkastella niiden arkkitehtuuria. Hecht-Nielsenin [22] mukaan arkkitehtuurilla tarkoitetaan neuroverkon formaalia matemaattista kuvausta. Arkkitehtuuri määrittää esimerkiksi neuroverkon neuronien ryhmittelyn ja niiden väliset yhteydet. Toisinaan puhutaan myös neuroverkon topologiasta. Neuroverkon kerroksien ja niiden välisten yhteyksien avulla on mahdollista määritellä tietynlaisia rakenteellisia vaatimuksia täyttäviä neuroverkkoarkkitehtuureja. Haykinin [21] mukaan voidaan tunnistaa pääasiassa kolmentyyppisiä neuroverkkoarkkitehtuureja: yksikerroksiset eteenpäinsyöttävät neuroverkot, monikerroksiset eteenpäinsyöttävät neuroverkot ja rekursiiviset neuroverkot.

*Eteenpäinsyöttävissä neuroverkoissa* (feedforward network) informaatio kulkee vain yhteen suuntaan neuroverkon läpi. Luonnollinen etenemissuunta on syötekerroksesta tuloskerrokseen. Neuronien tulokset voivat toimia ainoastaan etenemissuunnassa seuraavien kerrosten syötteinä. Eteenpäinsyöttävät neuroverkot voivat olla joko *yksikerroksisia* (single-layer) tai *monikerroksisia* (multi-layer). Yksikerroksinen neuroverkko koostuu syötekerroksen lisäksi ainoastaan yhdestä laskennallisesta kerroksesta, joka toimii samalla verkon tuloskerroksena. Monikerroksisessa neuroverkossa voi olla useampia laskennallisia kerroksia. Niitä ovat verkon tuloskerros sekä yksi tai useampi piilokerros.

Neuroverkkoa kutsutaan *täysin yhdistetyksi* (fully connected), jos jokaisen kerroksen kaikki neuronit on yhdistetty jokaiseen viereisen kerroksen neuroniin. Mikäli yksikin tämänkaltainen yhteys puuttuu, kutsutaan verkkoa *osittain yhdistetyksi* (partially connected).

*Rekursiiviset neuroverkot* (recurrent network) eroavat edellä kuvatuista eteenpäinsyöttävistä neuroverkoista siten, että niissä voi olla eteenpäinsyöttävien yhteyksien lisäksi taaksepäin johtavia yhteyksiä. Toisin sanoen neuronien tulokset voivat toimia myös edellisten kerrosten syötteinä. Lisäksi itseissilmukat ovat sallittuja; silloin neuroneista voi olla suora takaisinkytkentä itseensä.

Yksikerroksinen perceptron on esimerkki yksikerroksisesta eteenpäinsyöttävästä neuroverkosta. Vastaavasti monikerrosperceptron on monikerroksinen eteenpäinsyöttävä neuroverkko. *Hopfieldin verkko* (Hopfield network) on esimerkki rekursiivisesta neuroverkosta. Näiden lisäksi on olemassa myös toisenlaisia neuroverkkoja, kuten esimerkiksi *radiaalikantafunktioverkkoja* (radial basis network) ja *itseorganisoituvia karttoja* (self-organizing map). Radiaalikantafunktioverkoissa sovelletaan erityisiä aktivaatiofunktioita. Itseorganisoituissa kartoissa neuronit on järjestetty hilamaisiksi rakenteiksi. Molempia neuroverkkotyyppäjä käsitellään esimerkiksi lähteissä [17], [21] ja [29]. Tämän tutkielman kannalta keskeisimpiä ovat erityisesti monikerrosperceptron-neuroverkot.

### 3.3 Neuroverkon oppiminen

Ihmisen oppimiskyky perustuu pitkälti aivojen plastisuuteen. Aivoissa plastisuus ilmenee uusien synaptisten yhteyksien muodostumisena ja olemassa olevien muuttumisena [21]. Myös neuroverkon oppiminen perustuu sen parametrien, toisin sanoen verkon synaptisten painojen ja siirtotermien, muuttumiseen sopivalla tavalla. Neuroverkon oppimisprosessi voidaan jakaa vaiheisiin, joita Haykin [21] kuvaa seuraavasti:

1. Neuroverkko kohdistuu ympäristöstä lähtöisin olevalle stimulaatiolle.
2. Stimulaation seurauksena neuroverkon parametreissa tapahtuu muutoksia.



3. Neuroverkossa tapahtuneiden muutosten johdosta se reagoi uudella tavalla ympäristöönsä.

Kyseinen kuvaus on hyvin korkeatasoinen. Siinä ei esimerkiksi huomioida neuroverkon oppimisympäristöä. Oppimisparadigmat määrittelevät, kuinka neuroverkko ja sen oppimisympäristö ovat vuorovaikutuksessa keskenään. Oppimisparadigmat voidaan luokitella sen perusteella, valvooko neuroverkon oppimista opettaja vai tapahtuuko oppiminen ilman valvontaa. Mikäli oppimiseen liittyy opettaja, puhutaan valvotusta oppimisesta. Jos oppimiseen ei liity opettajaa, kyseessä on joko vahvistettu tai valvoton oppiminen.

Haykinin [21] mukaan *Valvotun oppimisen* (supervised learning) tavoitteena on saada neuroverkko matkimaan opettajaansa. Opettajan läsnäolo mallinnetaan käytännössä syöte ja tulos -esimerkkiparien avulla. Jokaiseen esimerkkipariin liittyvä tulososa vastaa opettajan antamaa tulosta kyseessä olevalle syötteelle. Neuroverkolle esitetään syötteitä ja sen tuloksia verrataan opettajan vastaaviin tuloksiin. Tavoitteena on etsiä sellaiset neuroverkon parametrien arvot, että neuroverkon tekemä virhe olisi riittävän pieni.

*Vahvistetussa oppimisessa* (reinforcement learning) ei ole varsinaista opettajaa, joka antaisi oikeita vastauksia. Oppiminen ohjautuu hieman erilaisen palautteen perusteella. Neuroverkolle esitetään erilaisia syötteitä ja sen jälkeen neuroverkon tulokset arvostellaan tai pisteytetään sopivasti. Oppimisen aikana ei voida tarkasti määrittää kuinka suuren virheen neuroverkko tekee, vaan järjestelmän on pääteltävä saamansa palautteen perusteella minkä suuntainen parametrien muuttaminen tuottaisi paremman tuloksen.

*Valvomaton oppiminen* (unsupervised learning) tapahtuu ilman minkäänlaista ohjausta. Neuroverkko luo oppimisen aikana itse kuvauksen syötteiden ja tulosten välille. Sitä käytetään esimerkiksi silloin, kun on saatavilla vain erilaisia syötteitä, mutta ei ole tietoa niihin liittyvistä tuloksista. Valvomatonta oppimista voidaan käyttää myös vahvistetun oppimisen sijasta, jos ei tiedetä edes sitä, kuinka tuloksia pitäisi arvostella. Vaikka valvomaton oppiminen onkin epätarkempaa kuin valvottu oppiminen, se on useimmiten kuitenkin nopeampaa ja se soveltuukin esimerkiksi vaativiin reaaliaikasovellyksiin [17]. Valvomatonta oppimista kutsutaan joissain yhteyksissä myös *klusteroinniksi* (clustering).

Neuroverkon ja oppimisympäristön välinen vuorovaikutus määräytyy edellä kuvattujen oppimisparadigmojen mukaan. Neuroverkon parametrien muuttuminen määräytyy vastaavasti sovellettavan oppimissäännön perusteella. Oppimissääntöjä on useita ja ne eroavat toisistaan pääasiassa siinä, kuinka ne muuttavat neuroverkon parametreja oppimisprosessin aikana. Erilaisia oppimissääntöjä ovat esimerkiksi Hebbin oppiminen, virheenkorjaava oppiminen, muistinvarainen oppiminen ja kilpaileva oppiminen. Niitä käsitellään yksityiskohtaisemmin kirjallisuudessa (katso esim. [21] ja [22]). Oppimissäännön tarkka algoritmien toteutus riip-

puu opetettavasta neuroverkosta. Myöhemmin esitellään monikerrosperceptronien opettamisessa käytetyn algoritmin toimintaa.

### 3.4 Neuroverkkojen sovelluksia

Neuroverkkoja voidaan käyttää useissa tekniikan ja tieteen sovelluksissa. Näitä ovat esimerkiksi sellaiset sovellukset, jotka käsittävät eri tyyppisiä mallinnus- tai luokittelutehtäviä. Sovellusten yksityiskohtainen luonne määräytyy kyseessä olevan sovellusalan ja varsinaisen tehtävän mukaan. Usein sovellusten kehittäminen edellyttää kyseisen sovellusalan hyvää tuntemusta [22]. Pelkkä neurolaskennan tuntemus onkin usein riittämätöntä oikeiden sovellusten suunnittelussa ja toteutuksessa.

Lukuisat neuroverkkosovellukset kertovat neuroverkkojen soveltuvuudesta eri tyyppisiin tehtäviin. Niiden tunteminen helpottaa neuroverkkojen toimintatavan ymmärtämistä. Toisinaan neuroverkkoja myös luokitellaan tehtävätyyppien perusteella (katso esim. [17]). Hecht-Nielsen [22] käsittelee joitain esimerkkisovelluksia yksityiskohtaisemmin. Eräs niistä on NETtalk-järjestelmä, joka on yksi kuuluisimmista neuroverkkosovelluksista. Haykin [21] käsittelee vastaavasti useita neuroverkkojen tehtävätyyppejä. Seuraavaksi esitellään joitain tyypillisimpiä tehtävätyyppejä.

*Hahmontunnistus* (pattern recognition) on luonnollista ihmisille. Ihmiset suorittavat hahmontunnistustehtäviä joka päivä: toisten ihmisten tunnistaminen kasvojen tai äänen perusteella, vaaratilanteiden tunnistaminen liikenteessä ja niin edelleen. Duda *et al.* [14] mukaan hahmontunnistuksella on ollut ratkaiseva asema ihmisen kyvyssä selviytyä. Sen johdosta ihmiselle on kehittynyt hienostunut hermostollinen ja kognitiivinen järjestelmä erityisesti hahmontunnistustehtävien suorittamiseen. Hahmontunnistuksessa tavoitteena on tarkasteltavien kohteiden eli *hahmojen* (pattern) luokittelu ennalta tunnettuihin *luokkiin* (class). Varsinainen luokittelu tapahtuu hahmoa kuvaavien *piirteiden* (feature) perusteella. Neuroverkkojen tapauksessa piirteet toimivat syöteinä ja luokittelu määräytyy neuroverkon tuloksen perusteella. Hahmontunnistus on laajempi kokonaisuus ja sitä käsitellään enemmän kirjallisuudessa (katso esim. [8], [14] ja [46]).

*Assosiatiivimuisti* (associative memory) on aivojen toiminnan kaltainen hajautettu muisti. Se on joko *autoassosiatiivinen* (autoassociative) tai *heteroassosiatiivinen* (heteroassociative). Autoassosiaatiossa neuroverkkoon tallennetaan hahmoja, jotka voidaan palauttaa myöhemmin antamalla verkolle syötteenä esimerkiksi kohinainen versio jostain tallennetusta hahmosta. Heteroassosiaatiossa neuroverkkoon tallennetaan hahmopareja, jotka eivät ole samanlaisia. Heteroassosiatiivinen verkko palauttaa syöteahmoa vastaavan parin.

*Funktion approksimoinnissa* (function approximation) tavoitteena on tietyn

funktion tai kuvauksen esittäminen neuroverkon avulla. Tavoitteena voi olla esimerkiksi määrittää jokin tuntematon kuvaus saatavilla olevien syöte ja tulos -esimerkkiparien avulla. Tässä tehtävässä neuroverkolle pyritään siis opettamaan funktion määrittäjäjoukon ja arvojoukon välinen yhteys. Myös käänteiskuvausten approksimointi on mahdollista [21].

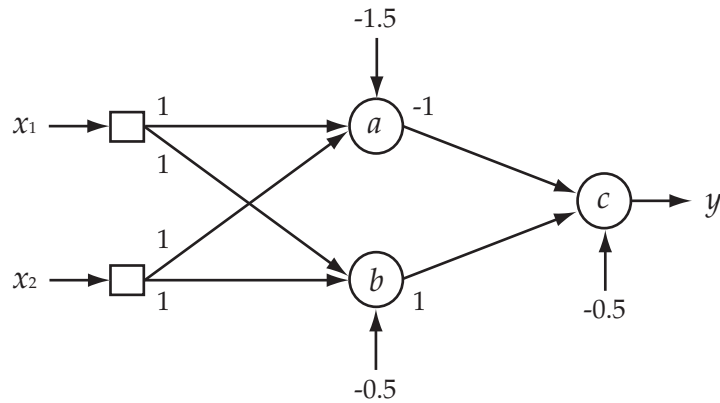
Neuroverkkoja voidaan käyttää myös erilaisten prosessien ohjaamiseen. Tehtävän aikana seurataan jonkin prosessin toimintaa ja pyritään ohjaamaan sen toimintaa toivottuun suuntaan. Neuroverkon syötteinä toimivat esimerkiksi prosessin toimintaa kuvaavat mitta-arvot. Tuloksina ovat vastaavasti prosessin toimintaan vaikuttavat ohjausarvot.

### 3.5 Monikerrosperceptron-neuroverkot

*Monikerrosperceptron* (multi-layer perceptron, MLP) on monikerroksinen eteenpäinsyöttävä neuroverkko. Siinä on syötekerroksen ja tuloskerroksen lisäksi yksi tai useampi piilokerros. Monikerrosperceptron perustuu Frank Rosenblattin 1950-luvulla kehittämään *perceptron*-neuroverkkoon, jossa aktivaatiofunktioina käytetään kynnsfunktioita. Yksinkertaisimmillaan perceptron koostuu vain yhdestä laskennallisesta neuronista. Yksi ainoa neuroni voi luokitella lineaarisesti erottuvia hahmoja vain kahteen luokkaan. Lisäämällä neuroneja samaan kerrokseen hahmoja voidaan luokitella useampiin luokkiin. Epälineaarisia ongelmia perceptronilla ei voida kuitenkaan ratkaista.

Kuuluisin esimerkki epälineaarisesta ongelmasta on niin sanottu XOR-ongelma, missä tavoitteena on ratkaista looginen *poissulkeva tai* (exclusive or) -operaatio. Yksikerroksisella perceptronilla ongelmaa ei voida ratkaista. Tämä Minskyn ja Papertin vuonna 1969 formaalista esittämä perceptronin heikkous lamautti neuroverkkotutkimuksen pitkäksi aikaa. XOR-ongelma voidaan ratkaista useampikerroksisella perceptronilla. Esimerkiksi kuvassa 3.3 on eräs kaksikerroksinen perceptron, joka ratkaisee kyseisen ongelman. Binäärimuuttujilla  $x_1$  ja  $x_2$  merkitään neuroverkon syötteitä ja muuttujalla  $y$  neuroverkon tulosta. Verkon laskennallisia neuroneja ovat piiloneuronit  $a$  ja  $b$  sekä tulosneuron  $c$ . Aktivaatiofunktioina käytetään unipolaarisia kynnsfunktioita. Painokertoimet on merkitty yhteyksien viereen. Siirtotermit on merkitty kuvaan vakiosyötteinä.

Tarkastellaan neuroverkon toimintaa, kun  $x_1 = 1$  ja  $x_2 = 0$ . Jo aikaisemmin todettiin, että neuroverkon syötekerroksessa ei tapahdu laskentaa. Syötearvot välitetään ainoastaan eteenpäin. Neuronin  $a$  kokonaisheräte on -0,5 ja neuronin  $b$  kokonaisheräte on 0,5. Kynnsfunktion määritelmän perusteella neuronien tulokset ovat vastaavasti 0 ja 1. Nämä arvot välittyvät edelleen neuronin  $c$  syötearvoiksi, jonka kokonaisherätteenä saadaan 0,5. Neuronin tulos on nyt 1, joka on samalla myös koko neuroverkon tulos  $y$ . Vastaavalla tavalla voidaan muodostaa



Kuva 3.3: Esimerkki XOR-ongelman ratkaisevasta kaksikerroksisesta perceptron-neuroverkosta.

Taulukko 3.1: Kuvan 3.3 neuroverkon tuloksia. Binäärimuuttujat  $x_1$  ja  $x_2$  ovat neuroverkon syötearvoja ja  $y$  on neuroverkon tulos. Tulos vastaa binäärimuuttujien  $x_1$  ja  $x_2$  välistä XOR-operaatiota.

$x_1$	$x_2$	$y$
1	0	1
1	1	0
0	0	0
0	1	1

taulukko 3.1, mikä vastaa binäärimuuttujien  $x_1$  ja  $x_2$  välistä XOR-operaatiota. XOR-ongelma voidaan ratkaista myös yhdellä kompleksiarvoisella neuronilla [32].

Haykinin [21] mukaan MLP-verkoilla on kolme tunnusomaista piirrettä. Ensiksi jokaisessa verkon neuronissa on epälineaarinen aktivaatiofunktio, joka on kaikkialla derivoituva (esim. logistinen sigmoidifunktio). Jos näin ei olisi, niin silloin neuroverkko redusoituisi yksikerroksisen perceptronin tasolle. Toiseksi verkossa on oltava yksi tai useampi piilokerros. Piilokerrosten neuronien avulla verkko kykenee oppimaan vaikeampia tehtäviä. Kolmanneksi MLP-verkoissa on paljon synaptisia yhteyksiä.

MLP-verkkojen teoreettinen tarkastelu saattaa olla vaikeaa useiden neuronien ja niiden välisten yhteyksien lukumäärän johdosta. Usein voidaankin kysyä kuinka suuri MLP-verkon tulee olla, jotta kyseessä oleva ongelma voitaisiin ratkaista. *Kolmogorovin teoreeman* (Kolmogorov's theorem) mukaan mikä tahansa jatkuva funktio voidaan esittää sellaisella eteenpäinsyöttävällä neuroverkolla, missä on yksi piilokerros. Teoreeman mukaan neuroverkossa, jossa on  $n$  syöteneuronia, on oltava  $2n + 1$  piiloneuronia. Syöteneuronien ja tulosneuronien lukumäärä määräytyy ap-

proksimoitavan funktion perusteella. Kolmogorovin teoreemalla on kuitenkin vähän käytännön merkitystä, koska se ei esimerkiksi kerro minkälaisia aktivaatiofunktioita verkossa tulisi käyttää [22]. Vastaus kysymykseen löytyykin useimmiten empiiristen kokeiden perusteella.

MLP-neuroverkkojen soveltuvuutta lukuisten ongelmien ratkaisemiseen lisää se, että niiden opettamiseen on olemassa tehokas *back-propagation* -algoritmi. Kyseinen algoritmi noudattaa niin sanotun *laskeutuvan gradientin* (gradient descent) menetelmää, missä tavoitteena on muuttaa painokertoimia siten, että verkon tekemä virhe pienenisi. Tämän perusteella algoritmi noudattaa virheenkorjaavan oppimisen periaatetta. Algoritmissa sovelletaan syöte ja tulos -esimerkkipareja, jolloin kyse on valvotusta oppimisesta.

Aluksi neuroverkon painokertoimet alustetaan pienillä satunnaisluvuilla. Tämän jälkeen algoritmin toimintaa voidaan kuvata kahden vaiheen avulla, joita toistetaan iteratiivisesti valvotun oppimisen mukaan. Ensimmäisessä vaiheessa neuroverkolle annetaan syötearvot, joiden perusteella se laskee jonkin tuloksen. Toisessa vaiheessa neuroverkon painokertoimia muutetaan virheenkorjaavan oppimisäännön mukaan. Algoritmin toiminnan aikana painokertoimia muutetaan toistuvasti jokaisella iteraatiolla, kunnes riittävän monta iteraatiota on takana tai neuroverkon virhe laskee ennalta määrätyn rajan alapuolelle. Näiden lisäksi myös toisenlaiset lopetusehdot ovat mahdollisia. Kirjallisuudessa opetuskertaa eli iteraatiota kutsutaan myös *epookiksi* (epoch).

Seuraavaksi esitetään tarkemmin, kuinka neuroverkon painokertoimia muutetaan back-propagation -algoritmissa. Tässä esityksessä mukaillaan Haykinin esitystä lähteessä [21]. Tarkastellaan painojen muuttamista mielivaltaisen neuronin  $j$  tapauksessa. Perusmuodossaan algoritmin  $n$ :llä iteraatiolla neuronista  $i$  neuroniin  $j$  kulkevan yhteyden painokerrointa muutetaan niin sanotun *delta-säännön* (delta rule) mukaan:

$$\Delta w_{ji}(n) = \eta g_j(n) y_i(n),$$

missä  $\eta$  on *oppimisnopeus* (learning rate),  $g_j(n)$  on neuronin  $j$  lokaali gradientti ja  $y_i(n)$  on neuronin  $i$  tulos, eli kyseisen syötekanavan heräte neuronille  $j$ . Tässä vaiheessa on syytä huomata, että  $i = 0$  vastaa neuronin  $j$  siirtotermiä, kuten aikaisemmin kaavassa (3.2). Jos neuroni  $j$  on tulosneuroni, niin silloin

$$g_j(n) = (y_j(n) - t_j(n)) f'_j(s_j(n)),$$

missä  $y_j(n)$  on neuronin tulos,  $t_j(n)$  on sen hetkinen tavoitetulos ja  $f'_j(s_j(n))$  on aktivaatiofunktion derivaatta, kun neuronin kokonaisheräte on  $s_j(n)$ . Jos neuroni  $j$  on piiloneuroni, niin silloin

$$g_j(n) = f'_j(s_j(n)) \sum_{k=0}^{K-1} g_k(n) w_{kj}(n),$$

missä  $f'_j(s_j(n))$  on kuten edellä,  $g_k$  on neuronin  $k$  lokaali gradientti ja  $w_{kj}(n)$  on neuronista  $j$  neuroniin  $k$  johtavan yhteyden painokerroin. Neuronit  $k$ , joita on  $K$  kappaletta, ovat neuronista  $j$  lähtevien yhteyksien päissä olevia neuroneja. Nyt seuraavalla iteraatiokierroksella  $n + 1$  neuronin  $j$  painokertoimet ovat

$$w_{ji}(n + 1) = w_{ji}(n) + \Delta w_{ji}(n).$$

Algoritmissa sovelletaan aktivaatiofunktioiden derivaattoja. Tämä huomio perustelee sen, miksi MLP:ssä on käytettävä derivoituvia aktivaatiofunktioita. Algoritmin toimintaa voidaan tehostaa käyttämällä *momenttitermiä* (momentum constant)  $\alpha$ . Sen lisäämisen jälkeen delta-sääntö saa muodon

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n - 1) + \eta g_j(n) y_i(n)$$

Ilman momenttitermiä neuroverkon oppiminen saattaa olla epästabiilia, kun käytetään suuria oppimisnopeuksia. Momenttitermillä verkon painokertoimien muutoksia voidaan tasoittaa siten, että suurempien oppimisnopeuksien soveltaminen on mahdollista.

Back-propagation -algoritmin kehittäminen oli merkkipaalu neurolaskennassa, sillä sen avulla voitiin opettaa MLP-verkkoja laskennallisesta näkökulmasta katsottuna edullisesti. Vaikka algoritmin ei voida väittää tarjoavan optimaalista ratkaisua kaikkiin ongelmiin, niin on se lopettanut neuroverkkoja kohtaan vallinneen pessimismin, jota ehkä Minskyn ja Papertin kirja aikanaan vahvisti [21]. Back-propagation -algoritmin toimintaa, tehostamista ja sen johtamista käsitellään yksityiskohtaisemmin kirjallisuudessa (katso esim. [3], [17], [21], [22] ja [29]).

## Luku 4

# Elektromyogrammi

Kehittyneet organismit muodostuvat elimistä, jotka hoitavat tiettyjä toimintoja. Elimet voivat muodostaa organismin suurempia osakokonaisuuksia ja osallistua jonkin fysiologisen prosessin toimintaan. Esimerkiksi ihmisen verenkiertoelimistö koostuu sydäimestä ja verisuonistosta. Sen pääasiallinen tehtävä on ravinteiden ja hapen kuljettaminen eri puolille kehoa. Elimistön toiminta synnyttää signaaleja, jotka voidaan useimmiten havaita sopivilla mittausvälineillä. Näitä signaaleja kutsutaan *biosignaaleiksi* (biomedical signal). Elävistä organismeista voidaan mitata useita erilaisia biosignaaleja. Ne voivat olla esimerkiksi sähköisiä, kemiallisia, akustisia, mekaanisia tai magneettisia signaaleja.

Sähköiset biosignaalit muodostavat erään biosignaalien joukon. Sähköisistä biosignaaleista tunnetuin on luultavasti *elektrokardiogrammi* (electrocardiogram), joka kuvaa sydänlihaksen sähköistä toimintaa. Aivojen sähköistä toimintaa tutkitaan *elektroenkefalogrammin* (electroencephalogram) eli aivosähkökäyrän avulla. Tutkielman kannalta merkittävässä asemassa on lihaksen sähköistä toimintaa kuvaava *elektromyogrammi* (electromyogram, EMG). Siksi tässä luvussa perehdytään ainoastaan elektromyogrammiin. Muita biosignaaleja käsitellään lähteissä [5] ja [38].

Aluksi tarkastellaan lihasten rakennetta, toimintaperiaatetta ja niiden toiminnasta syntyvää sähköistä aktiviteettia. Tutkimuksen aikana suoritettiin EMG-mittauksia, minkä johdosta on olennaista käsitellä EMG:n mittaamiseen liittyviä aiheita. Lopuksi esitellään joitain EMG:n kvantifiointimenetelmiä ja muutamia EMG:n sovelluksia.

### 4.1 Lihasten rakenne ja toiminta

Lihakset ovat tärkeä osa elimistöä. Ne mahdollistavat liikkumisen ja sisäelinten lihakset vastaavat useista tärkeistä kehon toiminnoista. Esimerkiksi pallealihas on

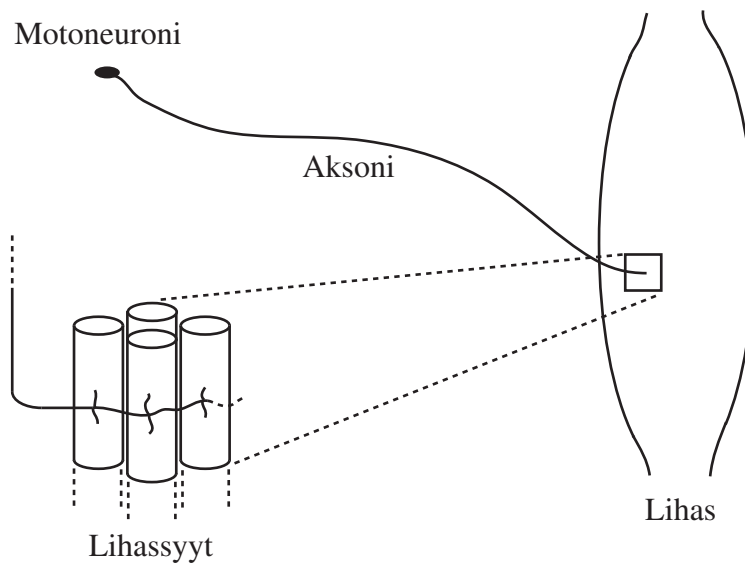
oleellinen osa hengityselimistöä ja sydänlihas vastaa veren kierrätyksestä kehos-  
sa. Lihaksisto koostuu kolmen tyyppisistä lihaksista. *Luurankolihakset* (skeletal  
muscle), jotka muodostuvat *poikkijuovaisesta lihaskudoksesta* (striated muscle),  
kiinnittyvät luihin, ja ne saavat aikaan kehon liikkeitä. Sisäelinten liikkeitä kehittä-  
vät vastaavasti *sileät lihakset* (smooth muscle). Kolmas lihastyypin on *sydänlihas*  
(cardiac muscle), jota esiintyy ainoastaan sydämessä. Pääasiassa vain luurankoli-  
hasten toimintaa on mahdollista ohjata tahdonalaisesti. Siksi poikkijuovaisesta li-  
haskudoksesta muodostuvat luurankolihakset ovatkin kiinnostavia tämän tutkiel-  
man kannalta. Tästä eteenpäin käsitellään vain luurankolihasrakennetta ja  
toimintaa.

Lihasten toiminta liikkeen ja voiman kehittäjinä on mahdollista niiden supis-  
tumiskyvyn ansiosta. Supistumiskyky perustuu lihasten rakenteeseen [4][5]. Lihas  
koostuu useista *lihassyistä* (muscle fiber), joiden halkaisijat ovat suuruudeltaan  
alle millimetrin luokkaa. Lihassyt koostuvat edelleen pienemmistä osista, *fibril-  
leista* (fibril), ja ne pääosin *aktiinista* (actin) ja *myosiinista* (myosin) muodostu-  
vista proteiinisauvoista. Proteiinisauvat ovat järjestäytyneet kerroksittain fibrillien  
suuntaisesti siten, että ne voivat liukua lomittain. Proteiinisauvojen lomittuminen  
aiheuttaa fibrillien lyhentymisen. Tämän seurauksena myös lihassyyn pituus lyhe-  
nee, jolloin sen sanotaan supistuvan.

Lihakset jakautuvat toiminnallisiin yksiköihin, joita kutsutaan *motorisiksi yk-  
siköiksi* (motor unit). Ne ovat pienimpiä lihasyksiköitä, joita voidaan aktivoida  
tahdonalaisesti [38]. Tässä esitetään lyhyt kuvaus motorisesta yksiköstä, mutta  
kirjallisuudessa niiden toimintaa ja toiminnan taustalla vaikuttavia sähkökemial-  
lisia reaktioita kuvataan tarkemmin (katso esim. [4] ja [47]). Motorisen yksikön  
(kuva 4.1) muodostavat *motoneuroni* (motoneuron, motor neuron), *viejähaarake*  
eli *aksoni* (axon) sekä kaikki sen hermottamat lihassyt. Motoneuronit sijaitse-  
vat tavallisesti selkäytimessä, mutta pään alueen lihasten tapauksessa ne sijait-  
sevat aivorungossa [4]. Motoneuronin aktivoituessa sen hermottamiin lihassyihin  
vapautuu välittäjäaineena toimivaa *asetyylkoliinia* (acetylcholine), mikä edelleen  
käynnistää sähkökemiallisia reaktioita lihassyissä. Niiden vaikutuksesta motorisen  
yksikön hermottamat lihassyt supistuvat. Lihaksen supistuminen on seurausta  
lihassyiden supistumisesta.

*Aktiopotentialilla* (action potential) tarkoitetaan yksittäisen solun toiminnas-  
ta syntyvää sähköistä signaalia. Se on kaikkien sähköisten biosignaalien peruskom-  
ponentti [38]. Sähköisten biosignaalien tapauksessa mitataan tavallisesti useiden  
solujen aktiopotentiaaleista muodostuvia jännitevaihteluja. Esimerkiksi motorisen  
yksikön toiminnasta syntyvä sähköinen signaali koostuu sen osien aktiopotentiaa-  
leista. Tätä signaalia kutsutaan *motorisen yksikön aktiopotentialiksi* (motor unit  
action potential, MUAP).



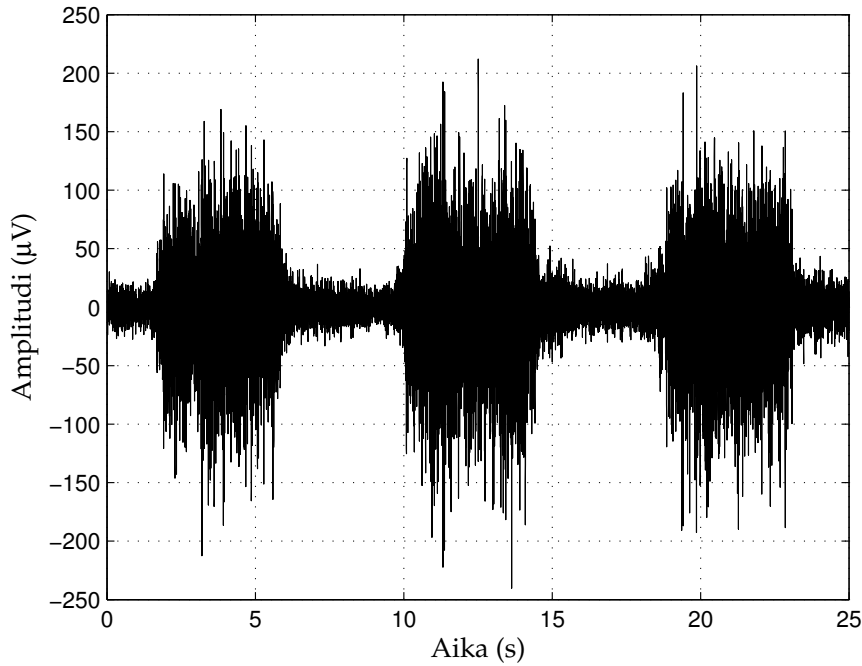


Kuva 4.1: Hahmotelma motorisesta yksiköstä (lähteen [4] sivulla 171 olevaa kuvaa mukaillen).

## 4.2 Sähköisen lihasaktiiviteetin mittaaminen

Lihaksen sähköisen aktiiviteetin mittaamista ja sen analysointia kutsutaan *elektromyografiaksi* (electromyography). Tässä kohdassa perehdytään EMG:n mittamiseen ja analysoinnissa käytettäviä menetelmiä käsitellään myöhemmin kohdassa 4.3. Perinteisin lihaksen sähköistä aktiiviteettia kuvaava signaali lienee *pinta-elektromyogrammi* (surface electromyogram). Se mitataan nimensä mukaisesti ihon pinnalta sopivilla pintaelektrodeilla. Kuvassa 4.2 on EMG-signaalinäyte, joka on mitattu pintaelektrodeilla *corrugator supercilii* -kasvolihaksen alueelta. Kuvan signaalissa on selvästi havaittavissa kolme lihaksen aktivaatiota. Lihaksesta voidaan mitata myös yksittäisen motorisen yksikön toimintaa ja tuloksena on jo aikaisemmin mainittu MUAP. Sen mittaamiseen käytetään tavallisesti neulaelektrodeja. Niiden avulla voidaan mitata lisäksi yksittäisen lihassyyn sähköistä toimintaa. Gazzoni *et al.* [18] esittävät menetelmän yksittäisten MUAP:ien havaitsemiseen myös pintaelektrodeilla mitatusta EMG:stä.

Elimistön ulkopuolelta tapahtuvaa mittausmenetelmää kutsutaan *kajoamattomaksi* eli *noninvasiiviseksi* (noninvasive). Elimistön sisälle ulottuvia mittausmenetelmiä kutsutaan vastaavasti *kajoaviksi* eli *invasiivisiksi* (invasive). Pintaelektrodeja käytettäessä kyseessä on noninvasiivinen mittaus ja neulaelektrodien tapauksessa invasiivinen mittaus. Vaikka invasiivisilla menetelmillä saadaan usein tarkempaa tietoa mittauskohteesta, liittyy niiden käyttöön Rangayyanin [38] mukaan riskejä. Lihaskudos saattaa nimittäin vahingoittua tai tulehtua ja mittaus voi olla kivulias.

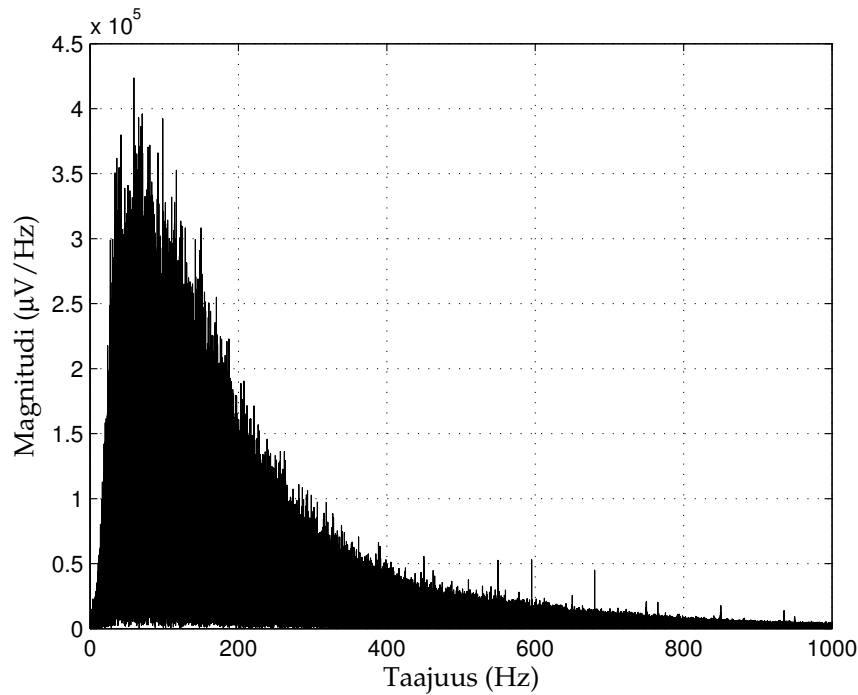


Kuva 4.2: EMG-signaalinäyte.

Koska noninvasiiviset menetelmät ovat käytännöllisempiä, käytettiin tutkimuksen aikana järjestetyissä EMG-mittauksissa pintaelektrodeja. Jatkossa perehdytäänkin ainoastaan pintaelektromyografiaan.

Pintaelektrodit ovat pohjimmiltaan ihon pinnalle kiinnitettäviä, johtimella varustettuja metallikappaleita. Elektrodien rakenne, koko ja muoto vaihtelevat (katso esim. [4], [9] ja [16]). Tavallisesti pintaelektrodit kiinnitetään ihoon kaksipuoleisella tarralla. Elektrodin ja ihon välissä käytetään elektrodipastaa tai -geeliä kontaktin parantamiseksi. Ihon pinnalla oleva rasva ja lika heikentävät oleellisesti sähköön virtaamista lihaksen ja elektrodien välillä. Siitä syystä ihon huolellinen puhdistaminen mittauskohteen alueelta ennen elektrodien kiinnittämistä on erityisen tärkeää. Iho puhdistetaan tavallisesti sopivalla liuotinaineella ja karhennetaan vanupuikolla tai hienorakeisella hiekkapaperilla. Tavoitteena on vähentää ihon *impedanssia* (impedance) eli vastusta noin 5–10 kilo-ohmiin ( $k\Omega$ ).

Pintaelektrodeilla mitatun EMG-signaalin amplitudin voimakkuus vaihtelee mikrovolttitasolta useaan millivolttiin ja taajuuskaista muutamasta hertsistä useaan tuhanteen hertsiin asti. Suurin osa signaalin taajuusinformatiosta keskittyy noin 20–200 Hz:n taajuuskaistalle. Tätä seikkaa tukee kuvan 4.3 taajuusspektri, joka on laskettu *corrugator supercilii* -kasvolihaksen alueelta mitatusta EMG-



Kuva 4.3: Tyypillinen EMG:n taajuusspektri.

signaalista. Pintaelektrodeilla mitattaessa suositeltava päästökaista on lähteestä riippuen 10–20 Hz:stä 200–1000 Hz:n.

Digitaalisia laitteita varten mitattu signaali on muunnettava digitaaliseen muotoon. Näytteenottotaajuus valitaan tavallisesti näytteenottoteoreeman mukaan. Fridlund ja Cacioppo [16] ohjeistavat kuitenkin käyttämään jopa 4–8 kertaa niin suurta näytteenottotaajuutta kuin signaalissa esiintyvä korkein taajuuskomponentti on. Tällöin voidaan välttää mahdollinen laskostuminen, mikä voi johtua laskostumisenestosuodattimen pidemmästä siirtymäkaistasta. Mittauksessa käytettävä vahvistus tulee myös valita sopivasti (katso esim. [11], [16] ja [47]). Vahvistuksen tulee olla sellainen, että signaalin arvoalue on vahvistuksen jälkeen rajoitettu. Liian suuri vahvistus voi nimittäin aiheuttaa signaalin leikkautumisen. Tämä seikka tulee huomioida esimerkiksi A/D-muunnoksen yhteydessä tehtävän näytearvojen kvantisoinnin johdosta.

EMG:n mittaamisessa tulee huomioida mahdolliset häiriötekijät. Häiriötekijöitä on niin kehon sisäpuolella kuin ulkopuolellakin. Tavallisimpia ovat kehon ulkopuolelta lähtöisin oleva taustakohina, kehon sisällä sijaitsevat muut fysiologiset signaalilähteet ja mittauksen aikana mahdollisesti syntyvät liikeartefaktat. Winter [47] luonnehtii ihmiskehoa johtimeksi, joka toimii antennin tavoin. Sähkömagneet-

tinen säteily onkin yksi tärkeimmistä häirölähteistä. Taustakohinalla tarkoitetaan radiosta, televisiosta, sähköjohdoista, sähkölampuista tai yleisesti mistä tahansa sähkölaitteesta lähtöisin olevaa sähkömagneettista säteilyä. Häiritsevintä on niin sanottu verkkovirtakohina, joka vaikuttaa 50/60 Hz:n taajuudella. Kyseisen häiriön suodatusta ei tavallisesti suositella, vaikka sopivan suodattimen suunnittelu olisi-kin helppoa. Suuri osa EMG:n taajuusinformaatiosta sijaitsee samalla taajuusalueella kuin verkkovirtakohina ja käytännössä suodatin vaimentaisi häiriön lisäksi myös EMG:n taajuuskomponentteja. Oikeaoppisella koehenkilöiden maadoituksella voidaan vaikuttaa verkkovirtakohinan esiintymiseen [4][16].

Mittausta häiritsevien fysiologisten signaalilähteiden vaimentamiseksi käytetään *kaksinapaista* (bipolar) elektrodien asettelua ja *erovahvistinta* (differential amplifier). Maaelektrodin lisäksi jokaisen mittaushaaran alueella käytetään kahta mittauselektrodia. Eroahvistin vahvistaa näiden mittauselektrodien mittaamien signaalien erotuksen. Molemmille elektrodeille näkyvä yhteinen signaali koostuu häirölähteiden muodostamista signaaleista. Eroahvistimessa kyseinen häiriösignaali vaimenee erotusoperaation johdosta. Menetelmällä saadaan vaimennettua fysiologisten häiriöiden lisäksi myös taustakohinaa. Täydellinen erotus on käytännössä mahdotonta ja suuretta, joka kuvaa kuinka hyvin tämä tapahtuu, kutsutaan *yhteisjännitevaimennussuhteeksi* (common-mode rejection ratio, CMRR) (katso esim. [9], [10] ja [47]).

*Liikeartefaktat* (movement artifacts) syntyvät pääasiassa ihon ja elektrodien johdinten liikkeistä. Liikkeistä aiheutuvan häiriön vaikutusalue rajoittuu kuitenkin matalille taajuuksille (noin 0–20 Hz). Sopivalla päästökaistan alarajan valinnalla häiriö saadaan vaimennettua. Vaihtoehtoisesti voidaan kokeilla johdinten vaihtamista tai mikrovahvistimella varustettujen elektrodien käyttämistä [47]. De Lucan [10] mukaan mittalaitteiden sisäinen kohina vaikuttaa myös mittaustulokseen. Kohina vaikuttaa 0 Hz:stä useaan tuhanteen hertsiin asti, eikä sen vaimentamiseksi voida tehdä juuri yhtään mitään. Ainoa keino häiriön vaimentamiseksi on käyttää korkealaatuisia elektronisia komponentteja sekä kehittyneitä laitteiden suunnittelu- ja valmistusmenetelmiä.

## 4.3 EMG:n kvantifointi

Pintaelektrodeilla mitattu elektromyogrammi on epästationaarinen signaali, joka koostuu useiden motoristen yksiköiden toiminnasta syntyvästä jännitevaihtelusta. Sen johdosta raa'an EMG:n perusteella on vaikea tehdä minkäänlaisia tulkintoja. Usein onkin parempi muuntaa signaali sellaiseen muotoon, että sitä on helpompi analysoida. EMG:n analysointiin sovelletaan erilaisia kvantifointimenetelmiä, jotta lihaksen toiminta voitaisiin hahmottaa helpommin. Menetelmien pääasiallisena tarkoituksena on muodostaa suhde EMG:n ja lihaksen toiminnallisen tilan välil-

lä. Tavoitteena saattaa olla esimerkiksi tiettyjen *motoristen tapahtumien* (motor event) havaitseminen.

Tavallisimpia kvantifointimenetelmiä ovat verhokäyrä, integrointi, RMS-arvo, nollakohtien lukumäärän laskeminen, huippujen lukumäärän laskeminen, käännösten lukumäärän laskeminen ja taajuusanalyysi. LeVeau ja Andersson [27] esittävät yhteenvedon erilaisista kvantifointimenetelmistä ja niiden käyttökohteista. Raaka EMG on kaikkien lihaksen toimintaa analysoivien menetelmien perusta. Vaikka se soveltuukin huonosti syvällisempien analyysien tekemiseen, voidaan sen avulla kuitenkin havaita häiritsevät artefaktat.

*Verhokäyrä* (linear envelope) kuvaa lihasaktiiviteetin profilia ajan funktiona. Ensin raakasignaali *tasasuunnataan* (rectify) ja sen jälkeen tasasuunnattu signaali alipäästösuodatetaan. Tuloksena saatu signaali seuraa tasasuunnatun signaalin vaihteluita. Tässä menetelmässä on kuitenkin huomioitava suotimesta syntyvä vaiheviive. Koska verhokäyrän laskennassa käytetty suodatin vaikuttaa tulossignaalin muotoon, on suodattimen ominaisuudet aina hyvä raportoida [27]. Verhokäyrä rinnastetaan toisinaan tasasuunnatun signaalin liukuvaan keskiarvoon. Jos suodattimen kertoimet valitaan sopivasti, niin tuloksena onkin tasasuunnatun signaalin liukuva keskiarvo.

Integroinnilla tarkoitetaan käyrän alla olevan pinta-alan laskemista. *Integroitu EMG* (integrated EMG, IEMG) kuvaa signaalin rajaaman alueen kokonaispinta-alan tiettyä ajanhetkenä. IEMG:n kasvunopeus vaihtelee signaalissa esiintyvän lihasaktiiviteetin mukaan; kasvunopeus on suurempi, kun signaalissa esiintyy voimakkaampaa lihasaktiiviteettiä. Koska IEMG kasvaa jatkuvasti, niin käytännössä se nollataan joko ajan tai jännitetason mukaan (katso esim. [27] ja [47]). *Aikanollauksessa* (time reset) IEMG:n nollaus tapahtuu kiinnitetyn ajanjakson välein. Voimakas lihasaktiiviteetti havaitaan tulossignaalin korkeiden huippujen lisääntymisenä. *Jännitetasonollauksessa* (voltage level reset) IEMG nollataan, kun se saavuttaa ennalta määrätyn jännitetason. Voimakas lihasaktiiviteetti havaitaan nollauskertojen lukumäärän kasvamisena.

*RMS-arvo* (Root Mean Square, RMS) mittaa signaalin sähköistä tehoa. Tarkalleen ottaen se on signaalin tehon neliöjuuri. Signaalin  $x(n)$  RMS-arvo määritellään kaavalla

$$RMS = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2},$$

missä  $N$  on signaalin pituus. Edellinen antaa koko signaalin RMS-arvon. Liukuva RMS-arvo määritellään kaavalla

$$RMS(n) = \sqrt{\frac{1}{L} \sum_{j=0}^{L-1} |x(n-j)|^2},$$

missä  $L$  on signaali-ikkunan pituus ja  $n$  saa arvot  $0, 1, \dots, N - 1$ . Liukuva RMS-arvo muistuttaa aikaisemmin esiteltyä verhokäyrää. LeVeau ja Anderssonin [27] mukaan RMS-arvo esittää matemaattisesti alkuperäistä signaalia kuitenkin jonkin verran paremmin.

Eräs tapa lihasaktiiviteetin seuraamiseen on laskea signaalissa esiintyvien ilmiöiden lukumääriä. *Nollakohtien lukumäärä* (zero crossing rate) kertoo, kuinka monta kertaa signaali leikkaa nollatason tietyn pituisen ajanjakson aikana. *Huippujen lukumäärä* (spike count) on signaalin positiivisten ja negatiivisten huippujen lukumäärä tietyllä ajanjaksolla. *Käännösten lukumäärä* (turn count) kuvaa vastaavasti sitä, kuinka usein signaali vaihtaa suuntaa tietyn pituisen ajanjakson aikana. Muiden muassa LeVeau ja Andersson [27] ja Rangayyan [38] käsittelevät näitä menetelmiä tarkemmin.

*Taajuusanalyysi* (spectral analysis) tuottaa signaalin energiajakauman taajuuksien funktiona. Lihaksen jatkuvan supistuksen seurauksena korkeat taajuuskomponentit heikkenevät ja matalat taajuuskomponentit vastaavasti voimistuvat. Taajuusspektrissä voidaankin havaita *mediaanitaajuuden* (median frequency) ja *keskiarvotaajuuden* (mean frequency) siirtyminen.

## 4.4 EMG:n sovelluksia

Edellisessä kohdassa esiteltiin joitain tyypillisimpiä kvantifointimenetelmiä, joita sovelletaan EMG:n yhteydessä. Vaikka suuri osa näistä menetelmistä tähtääkin lihaksen toiminnallisen tilan hahmottamiseen, voidaan EMG:tä soveltaa siitä huolimatta useissa erilaisissa sovelluksissa. Lääketieteessä EMG-signaaleja käytetään erilaisten neuromuskulaaristen sairauksien, kuten *neuropatian* (neuropathy) ja *myopatian* (myopathy), lihasten väsymisen, lihaskipujen ja epänormaalien lihastoimintojen tutkimisessa (katso esim. [5], [37] ja [38]). EMG-signaaleja käytetään myös psykofysiologisissa tutkimuksissa. Esimerkkinä mainittakoon kasvolihasista mitatun EMG-signaalin soveltaminen ihmisen tunneperäisen prosessoinnin tunnistamiseen (katso esim. [4] ja [25]).

Varsinkin EMG:n soveltumista osaksi erilaisia käyttöliittymiä on tutkittu. Eräs tutkimuskohde on keinoitekoisten proteesien ohjaaminen EMG-signaalien avulla (tai tähän tarkoitettujen menetelmien tutkiminen), mitä muun muassa Reischl *et al.* [39] sekä Soares *et al.* [42] ovat tutkineet. Elektromyogrammia voidaan käyttää hyväksi myös pyörätuolia ohjatessa. Esimerkkinä tämän kaltaisesta käyttöliittymästä mainittakoon Felzerin ja Freislebenin [15] kehittämä HaWCoS-järjestelmä. Ei pidä kuitenkaan unohtaa tutkimusta ihmisen ja tietokoneen välisistä käyttöliittymistä. EMG-signaalien hyödyntämistä on tutkittu myös näiden käyttöliittymien osalta (katso esim. [2], [6] ja [34]). Tutkimusten eräänä päämääränä on parantaa fyysisesti rajoittuneiden henkilöiden mahdollisuuksia käyttää tietokoneita. Luon-

nollisesti tällainen järjestelmä helpottaisi myös fyysisesti rajoittuneiden henkilöiden toimimista tietoyhteiskunnassa ja mahdollisesti parantaisi heidän elämänlaatuaan.

Myös neuroverkkoja on käytetty erilaisissa EMG:n sovelluksissa. Tässä mainitaan esimerkkinä muutamia EMG:n neuroverkkosovelluksia. Abel *et al.* [1] käyttivät neuroverkkoja lihassairauksien luokitteluun EMG:n perusteella. Liu *et al.* [28] sovelsivat neuroverkkoa EMG:n perusteella tapahtuvaan lihaksen jännitysvoiman ennustamiseen. Pah *et al.* [35] käyttivät neuroverkkoa EMG:n luokitteluun lihasten väsymisen perusteella. Dipietro *et al.* [13] käyttivät neuroverkkoa yläraajojen liikkeiden tunnistamiseen. Tässä tutkielmassa neuroverkkoja käytetään lihaksen sähköisen toiminnan muutosten havaitsemiseen.

## Luku 5

# Neuroverkkopohjainen ilmaisinjärjestelmä

Tutkielman tarkoituksena on tarkastella, kuinka neuroverkot soveltuvat lihaksen sähköisen toiminnan muutosten havaitsemiseen EMG-signaalin perusteella. Erityisesti käsitellään muutoksia, jotka kertovat tahdonalaisista lihastoiminnoista. Tehtävää varten suunniteltiin neuroverkkopohjainen ilmaisinjärjestelmä, jonka toimintaa testattiin kokeellisesti. Tutkimusaineiston muodostivat kolmenkymmenen koehenkilön kasvojen lihaksista mitatut EMG-signaalit. Ne mitattiin tämän tutkimuksen aikana. Ilmaisinjärjestelmä suunniteltiin reaaliaikainen toiminta huomioiden, mutta tässä vaiheessa tutkimus keskittyi järjestelmän toimintaan ei-reaaliaikaisessa ympäristössä. Erityisesti perehdyttiin MLP-neuroverkkojen toimintaan osana kyseistä järjestelmää. Tämän lisäksi tarkasteltiin sitä, kuinka digitaalisten esikäsittelymenetelmien soveltaminen vaikuttaa neuroverkkojen suorituskykyyn vaihtelevissa kohinaolosuhteissa.

### 5.1 Ilmaisinjärjestelmän suunnitelma

Lihaskiviteetin muutosten havaitsemisessa on pohjimmiltaan kysymys tietynlaisten motoristen tapahtumien havaitsemisesta. Tärkeitä tapahtumia ovat erityisesti lihasaktiiviteetin alkaminen ja päättymisen. EMG:ssä esiintyvät muutokset viestivät lihaksen sähköisen toiminnan vaihteluista ja liittyvät olennaisesti lihaksen motorisiin tapahtumiin. Staude ja Wolf [43] esittävät formaalin mallin motoristen tapahtumien havaitsemiselle. Sen mukaan tietokoneistettu tapahtumien havaitseminen voidaan jakaa kolmeen vaiheeseen:

1. Signaalin laadun parantamiseksi se esikäsitellään jollain sopivalla menetelmällä.



2. Varsinaisesta tapahtumien havaitsemisesta vastaa ilmaisinyksikkö. Se koostuu testifunktiosta ja päättelysäännöstä. Signaalin näytearvoista lasketaan testifunktion avulla jokin tunnusarvo. Päättelysääntö tekee varsinaisen päätöksen tapahtuman esiintymisestä tunnusarvon perusteella.
3. Jälkikäsitteilyn tavoitteena on epäluotettavien tapahtumien hylkääminen ja luotettavien tapahtumien esiintymishetkien määrittäminen.

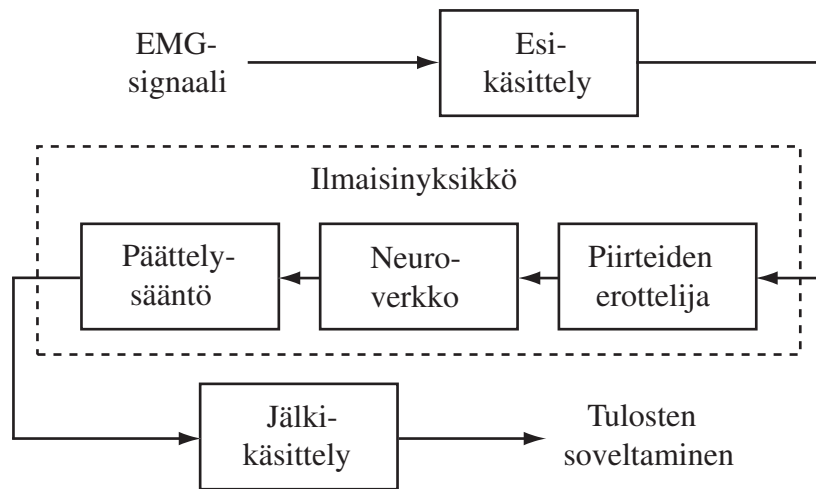
Staude ja Wolf huomauttavat, että malli heijastelee tietokoneistetun tapahtumien havaitsemisen formaaleja piirteitä. Sitä ja oikeita sovelluksia vertailtaessa esiintyy todennäköisesti joitain eroavuuksia. Edelliset vaiheet ovat siitä huolimatta yleisiä kaikkien kokoonpanojen yhteydessä.

Motoristen tapahtumien havaitsemiseen käytetään useita menetelmiä. Menetelmien lukumäärän perusteella voi päätellä, ettei ole olemassa yhtä yleisesti hyväksyttyä ja käytettyä menetelmää. Siitä huolimatta useimmat menetelmät noudattavat ainakin jossain määrin Stauden ja Wolfin mallia. Ne perustuvat useimmiten jonkinlaisen testifunktion laskemiseen, minkä tulosta verrataan sopivasti määritellyyn kynnysarvoon. Eri tyyppisiä menetelmiä käsitellään lähteissä [23], [26], [31], [43] ja [44]. Menetelmät eroavat toisistaan pääasiassa siinä, kuinka testifunktio ja kynnysarvo määritellään.

Stauden ja Wolfin malli toimii pohjana tutkielmassa käytetyn ilmaisinjärjestelmän suunnittelussa. Tavoitteena oli suunnitella järjestelmä, joka rakentuisi vastaavalla tavalla tiettyjä tehtäviä suorittavista komponenteista. Tämä on perusteltua, koska suunnitteluvaiheessa ei voida välttämättä tietää kuinka lopullinen järjestelmä tulee rakentumaan. Lähestymistavan ansiosta ilmaisinjärjestelmän toimintaa on helppo testata myös vaihtoehtoisilla komponenteilla. Niitä ovat esimerkiksi vaihtoehtoiset esikäsitteilymenetelmät

Koska tutkielman tarkoituksena on tarkastella neuroverkkojen soveltumista kyseessä olevaan tehtävään, on neuroverkkopohjaisuus keskeisimpiä vaatimuksia joita järjestelmän on täytettävä. Eräs neuroverkkojen tärkeimpiä ominaisuuksia on yleistämiskyky. Ominaisuus on hyödyksi motoristen tapahtumien havaitsemisessa, koska yleistämiskykyistä järjestelmää on helpompi käyttää eri henkilöiltä mitattujen EMG-signaalien yhteydessä. Lisäksi tehtävä voidaan käsittää hahmontunnistuongelmana, missä tarkasteltavaa signaalia pyritään luokittelemaan sen piirteiden perusteella. Hahmontunnistus onkin eräs niistä tehtävistä, joihin neuroverkot soveltuvat erityisen hyvin.

Ilmaisinjärjestelmä koostuu Stauden ja Wolfin mallin tavoin kolmesta komponentista: esikäsitteily, ilmaisinyksikkö ja jälkikäsitteily. Ilmaisinyksikkö koostuu piirteiden erottelijasta, neuroverkosta ja päättelysäännöstä. Järjestelmän rakenne on esitetty yleistasolla kuvassa 5.1. Seuraavaksi tarkastellaan komponenttien tehtäviä yksityiskohtaisemmin.



Kuva 5.1: Ilmaisinyksikön rakenne yleistasolla.

Esikäsittelyvaiheen tarkoituksena on EMG-signaalin laadun parantaminen. Signaalin esikäsittelyssä voidaan käyttää periaatteessa mitä tahansa lineaarista tai epälineaarista DSP:n menetelmää. Tavallisesti EMG:n yhteydessä käytetään kais-tanpäästösuodatinta, joka vaimentaa tietyn taajuuskaistan ulkopuolisia taajuus-komponentteja. Tavoitteena on signaalissa esiintyvän kohinan tai haitallisten ar-tefaktojen poistaminen, koska ne voisivat vaikuttaa järjestelmän toimintaan siten, että se tuottaisi virheellisiä tuloksia. Ilmaisinyksikön kannalta esikäsittelyssä voidaan käyttää mitä tahansa menetelmää, kunhan se soveltuu EMG-signaalin esikäsittelyyn.

Ilmaisinyksikkö on keskeisin komponentti, koska sen tehtävänä on varsinainen tapahtumien havaitseminen. Ilmaisinyksikkö koostuu piirteiden erottelijasta, neuroverkosta ja päätelysäännöstä. Mikäli ilmaisinyksikön rakennetta vertaillaan Stauden ja Wolfin malliin, vastaa neuroverkko testifunktiota. Tässä järjestelmäs-sä testifunktion arvo lasketaan kiinnitetyn pituisen signaali-ikkunan perusteella. Esikäsitelty signaali kulkee tavallaan tämän ikkunan läpi ja testifunktion arvo las-ketaan tietyin väliajoin.

Piirteiden erottelija segmentoi signaalin ja muodostaa kaksi syötearvoa neuro-verkolle. Piirteiden muodostuksessa käytetään kohdassa 4.3 esiteltyä RMS-arvoa, joka liittyy signaalin sähköiseen tehoon. RMS-arvot lasketaan äärellisen pituisen signaali-ikkunan puolikkaille ja nämä arvot normalisoidaan koko ikkunan alueel-ta lasketulla RMS-arvolla. Tästä seuraa piirreavaruuden invarianttisuus signaalin voimakkuuden suhteen. Invarianttisuuden aikaansaamiseksi voidaan käyttää myös muita menetelmiä (katso esim. [3] ja [21]). Haykinin [21] mukaan invariantin piir-reavaruuden käyttäminen saattaa kuitenkin olla paras menetelmä neuroverkkojen

kannalta.

Tarkastellaan piirreavaruuden invarianttisuutta tarkemmin. Edellä mainittiin, että syötearvot lasketaan äärellisen pituisen signaali-ikkunan alueelta. Merkitään signaali-ikkunaa äärellisen pituisen signaalin  $x(n)$  avulla. Olkoon sen pituus parillinen kokonaisluku  $N$ . Määritellään signaali-ikkunan RMS-arvo kaavalla

$$RMS_w = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2}.$$

Signaali-ikkunan ensimmäisen puolikkaan RMS-arvo määritellään kaavalla

$$RMS_1 = \sqrt{\frac{1}{N/2} \sum_{n=0}^{N/2-1} |x(n)|^2}$$

ja jälkimmäisen puolikkaan RMS-arvo vastaavasti kaavalla

$$RMS_2 = \sqrt{\frac{1}{N/2} \sum_{n=N/2}^{N-1} |x(n)|^2}.$$

Oletetaan, että signaali-ikkunan energia keskittyy ensimmäisen puolikkaan alueelle. Nyt ensimmäistä puolikasta vastaava normalisoitu RMS-arvo

$$NRMS_1 = \frac{RMS_1}{RMS_w} = \frac{\sqrt{\frac{1}{N/2} \sum_{n=0}^{N/2-1} |x(n)|^2}}{\sqrt{\frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2}} \quad (5.1)$$

saa maksimiarvonsa. Koska energia keskittyy vain signaali-ikkunan alkupuolelle, on jälkimmäistä puolikasta vastaava normalisoitu RMS-arvo

$$NRMS_2 = \frac{RMS_2}{RMS_w} = \frac{0}{\sqrt{\frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2}} = 0.$$

Edellä oletettiin, että kaikki energia kohdistuu signaali-ikkunan alkupuolelle. Silloin kaavan (5.1) osoittajan ja nimittäjän summausekkeet ovat yhtäsuuria. Havainnosta seuraa, että

$$NRMS_1 = \frac{\sqrt{\frac{1}{N/2} \sum_{n=0}^{N/2-1} |x(n)|^2}}{\sqrt{\frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2}} = \frac{\sqrt{\frac{1}{N/2}}}{\sqrt{\frac{1}{N}}} \cdot \underbrace{\frac{\sqrt{\sum_{n=0}^{N/2-1} |x(n)|^2}}{\sqrt{\sum_{n=0}^{N-1} |x(n)|^2}}}_{=1} = \sqrt{2}.$$

Edellisen perusteella normalisoidut RMS-arvot kuuluvat suljetulle reaaliarvovälille  $[0, \sqrt{2}]$ . Koska syötearvoina käytetään normalisoituja RMS-arvoja, niin ilmaisimen toiminta ei ole riippuvainen absoluuttisista arvoista. Todellisuudessa koko signaali-ikkunan alueella on energiaa, eivätkä normalisoidut RMS-arvot voi silloin saada määritettyjä ääriarvoja.

Neuroverkon tehtävänä on muodostaa kuvaus signaali-ikkunoiden piirearvoilta kolmelle luokalle: lihasaktiviteetin alkaminen, lihasaktiviteetin päättymisen sekä tasannekohta. Jälkimmäisen luokan signaali-ikkunoissa ei ole havaittavissa merkittäviä motorisia tapahtumia. Neuroverkon syötekerroksen tulee olla sopivan kokoinen, eli vastaanottaa kaksi syötearvoa. Tuloskerroksessa on oltava kolme tulosneuria, joista kukin vastaa yhtä edellä mainituista luokista. Tulosneuroneiden arvoalueiden on rajoitettava suljetulle reaaliarvovälille  $[0, 1]$ . Sen tähden tulosneuroneissa on käytettävä sopivia (esim. sigmoidisia) aktivaatiofunktioita. Muita rakenteellisia vaatimuksia neuroverkolle ei aseteta. Neuroverkon arkkitehtuuri voidaan valita useista vaihtoehdoista.

Neuroverkon tulos välitetään päättelysäännölle, jonka tehtävänä on päättää mihin luokkaan signaali-ikkuna kuuluu. Tässä päättelysääntönä käytetään yksinkertaista vertailijaa, joka valitsee signaali-ikkunan luokaksi sen, jota vastaavan tulosneuroniin arvo on suurin. Tämä on mahdollista, koska logistisen aktivaatiofunktion arvot voitiin tulkita todennäköisyyksinä. Toisin sanoen ikkunan luokitteluksi valitaan sellainen luokka, joka on neuroverkon tulosten perusteella todennäköisin. Haykinin [21] mukaan kyseinen päättelysääntö on parempi verrattuna kiinteän kynnyksarvon menetelmään, jossa kynnyksen ylittävät arvot määräävät luokittelun. Tämä johtuu siitä, että kiinteän kynnyksarvon menetelmällä on mahdollista luokitella sama alkio useampaan luokkaan. Maksimiarvoon perustuvassa luokittelussa tämä on epätodennäköisempää.

Ilmaisinyksikön tuottama luokittelu välittyy edelleen jälkikäsitteilyjälle, jonka tehtävänä on epäluotettavien luokittelujen hylkääminen ja tapahtuman esiintymishetken approksimointi. Epäkelpoksi luokitteluksi voidaan määritellä esimerkiksi sellainen tapahtuma, jota edeltää samantyyppinen tapahtuma. Tämä riippuu kuitenkin tilanteesta ja toisinaan epäkelvon luokittelun määritelmä voikin vaihdella. Arvio tapahtuman esiintymishetkelle voidaan määrittää esimerkiksi signaali-ikkunan puolesta välissä olevan näytearvon indeksin avulla. Jälkikäsitteilyssä onkin signaali-ikkunan luokittelun ja pituuden lisäksi tiedettävä esimerkiksi signaali-ikkunan uusimman näytearvon indeksi.

Tutkielmassa käytetyn ilmaisinjärjestelmän toiminta voidaan kiteyttää seuraaviin vaiheisiin:

1. Ilmaisinjärjestelmä saa syötteenä EMG-signaalin, joka esikäsitellään jollain sopivalla menetelmällä.
2. Ilmaisinyksikössä tapahtuu piirteiden erottelu ja syötearvojen muodostami-

nen, neuroverkolla laskeminen sekä lopuksi päättelysäännön soveltaminen signaali-ikkunan luokittelun määrittämiseksi.

3. Jälkikäsittelevä vaiheessa hylätään epäluotettavat luokittelut ja mahdollisesti approksimoidaan tarkemmin tapahtuman esiintymishetkeä.

Tämän jälkeen toiminta riippuu pitkälti käytännön sovelluksesta, jonka osana ilmaisinyjärjestelmä toimii. Ilmaisinyjärjestelmän toiminta edellyttää joidenkin parametrien määrittämisen. Näitä ovat ikkunan pituus, siirtymän pituus ja päättelysäännön toimintaan vaikuttava herkistystermin.

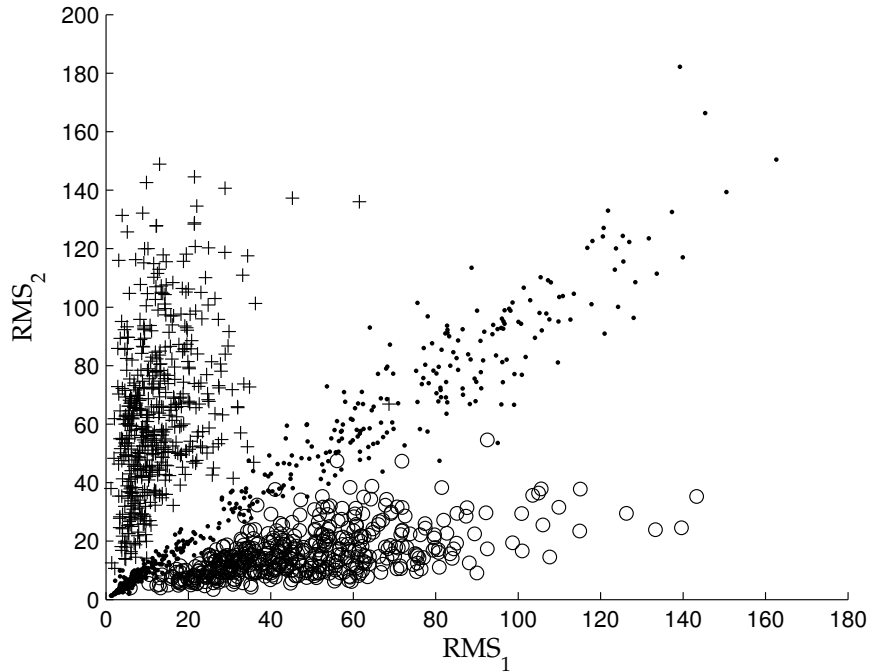
Ikkunan pituus on parillinen kokonaisluku, joka määrittää kuinka monen näytearvon perusteella neuroverkon syötearvot muodostetaan. Tämä parametri on hyvä valita siten, että se vastaa ajallisesti neuroverkon kouluttamisessa käytettyä ikkunan pituutta. Silloin ilmaisinyjärjestelmä etsii tapahtumia, jotka esiintyvät vastaavan pituisen aikavälin alueella. Ikkunan pituus voidaanakin kiinnittää samaksi kuin opetuksen aikana, vaikka RMS-arvojen laskennassa tapahtuukin normalisointi signaalin pituuden suhteen. Näytteenottotaajuus vaikuttaa olennaisesti ikkunan pituuteen. Onkin syytä huomioda, että näytteenottotaajuuden muuttuessa myös ikkunan pituutta on muutettava sopivassa suhteessa.

Siirtymän pituus määrittää kuinka pitkän ajan välein tapahtumien havaitsemisen suoritetaan. Käytännössä siirtymän pituus on järkevää valita yhden näytteen ja ikkunan pituuden väliltä. Vaikka ääriarvot ovatkin käytettävissä, löytyy sopiva siirtymän pituus todennäköisemmin niiden väliltä. Jos nimittäin siirtymä on yhden näytearvon pituinen, tulee jälkikäsittelevän kyetä tehokkaasti hylkäämään epäkelvoja tapahtumia. Toisaalta kun siirtymä on liian pitkä, saattaa osa tapahtumista jäädä havaitsematta. Siirtymän pituuden valintaan vaikuttavat myös käytännön sovellus ja sille asetetut vaatimukset.

Ilmaisinyksikön toimintaa voidaan säätää herkistystermin avulla. Se käsittää kolme arvoa suljetulta reaaliarvoväliltä  $[0, 1]$ . Herkistystermin vaikutus neuroverkon ja päättelysäännön välillä. Jokaisen tulosneuronin arvo kerrotaan vastaavalla herkistystermin arvolla. Näin voidaan vaikuttaa siihen, kuinka herkästi päättelysääntö valitsee tietyn luokittelun. Herkistystermin vaikutusta ei kuitenkaan tarkastella tässä tutkielmassa.

Esitelty suunnitelma kattoi kokonaisen ilmaisinyjärjestelmän, vaikka tutkielmassa tarkastellaan pääasiassa neuroverkkojen soveltumista lihaksen sähköisen toiminnan muutosten havaitsemiseen. Ilmaisinyjärjestelmän tarkoituksena onkin luoda jonkinlainen kehysovellus neuroverkolle. Se helpottaa omalta osaltaan neuroverkon toiminnan tarkoituksen ymmärtämistä.

Lopuksi havainnollistetaan RMS-arvojen normalisoinnin vaikutusta. Kuvassa 5.2 on signaali-ikkunoista laskettujen RMS-arvojen hajontakuva ennen normalisointia. Symbolilla + merkitään lihasaktiiviteetin alkamista, symbolilla  $\cdot$  tasannekohtaa ja symbolilla  $\circ$  lihasaktiiviteetin päättymistä. Kuvassa 5.3 on vastaavien

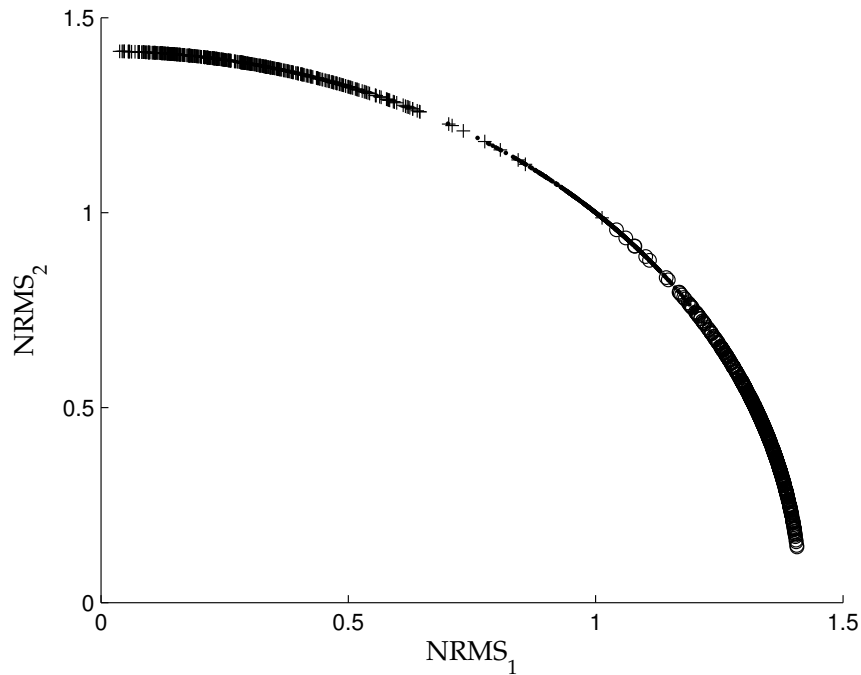


Kuva 5.2: RMS-arvojen hajontakuvio ennen normalisointia. Symbolilla + merkitään lihasaktiiviteetin alkamista, symbolilla  $\cdot$  tasannekohtaa ja symbolilla  $\circ$  lihasaktiiviteetin päättymistä.

RMS-arvojen hajontakuvio normalisoinnin jälkeen. Normalisoinnin vaikutus havaitaan selvästi kuvia vertailemalla. Ensinnäkin normalisoidut arvot rajoittuvat suljetulle reaaliarvovälille  $[0, \sqrt{2}]$ . Toisin sanoen piirreavaruus on invariantti signaalin voimakkuuden suhteen, minkä johdosta invarianttisuutta ei tarvitse rakentaa neuroverkkoon. Toiseksi ongelma yksinkertaistuu, kun absoluuttisista arvoista luovutaan ja siirrytään suhteellisiin arvoihin. Silloin ongelman ratkaisemiseen voidaan soveltaa myös pienempiä neuroverkkoja.

## 5.2 Tutkimusaineiston kerääminen

Tutkimusaineiston muodostivat kolmenkymmenen koehenkilön kasvojen lihaksista mitatut EMG-signaalit. Mittauskohteina olivat *corrugator supercilii* ja *zygomaticus major* -kasvolihakset. *Corrugator supercilii* -kasvolihakset sijaitsevat silmien sisäkulmien yläpuolella. Ne aktivoituvat esimerkiksi silloin, kun kulmakarvoja vedetään alaspäin. Kansankielisemmin puhutaan otsan rypistämisestä. *Zygomaticus major* -kasvolihakset sijaitsevat poskien alueella ja ne aktivoituvat esimerkiksi hy-



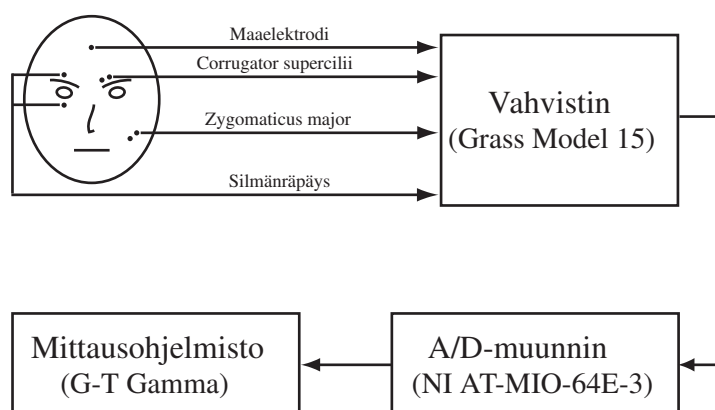
Kuva 5.3: RMS-arvojen hajontakuvio normalisoinnin jälkeen. Symbolilla + merkitään lihasaktiiviteetin alkamista, symbolilla · tasannekohtaa ja symbolilla o lihasaktiiviteetin päättymistä.

myiltäessä. Koehenkilöitä oli helppo ohjeistaa, koska hymyileminen ja otsan rypistäminen ovat tyypillisiä kasvojen liikkeitä.

Edellisten mittauskohteiden lisäksi mitattiin silmänräpäysaktiiviteetti. Aluksi sitä ajateltiin käyttää silmien räpäyksistä syntyvien artefaktien poistamiseen, jotka mahdollisesti esiintyisivät muissa mittauskanavissa. Koska silmänräpäykset eivät oleellisesti haitanneet muita mittauskanavia, ei silmänräpäysaktiiviteettiä käytetty tähän tarkoitukseen.

Mittauksissa käytettiin pyöreitä hopea/hopeakloridi (Ag/AgCl) -pintaelektrodeja. Ne kiinnitettiin kaksipuoleisilla tarroilla mitattavien kasvolihasten alueille. Kiinnityskohdat puhdistettiin ennen elektrodien kiinnitystä, jotta ihon impedanssi saataisiin riittävän alhaiseksi. Näiden mittauksen aikana tavoitteena oli saavuttaa alle 10 kΩ:n impedanssiarvo. Kiinnityskohdat puhdistettiin alkoholipitoisella puhdistusaineella ja ihoa karhennettiin elektrodipastaan kastetulla vanupuikolla. Elektrodipasta soveltui tähän tehtävään rakeisuutensa puolesta. Samaa elektrodipastaa käytettiin myös elektrolyytinä elektrodien ja ihon välissä.

Mittauksissa sovellettiin kaksinapaista elektrodien asettelua ja erovahvistusta.



Kuva 5.4: Mittausasetelma.

Kaksinapaisen asettelun mukaan jokaiseen mittauskohteeseen kiinnitettiin kaksi elektrodia vierekkäin. Silmänräpäysaktiviteetin tapauksessa elektrodit kiinnitettiin kulmakarvan yläpuolelle ja alaluomen alapuolelle. Erovahvistuksella pyrittiin vaimentamaan mahdollisia häiriöitä jo mittausvaiheessa. Sen edellyttämä maaelektrodi kiinnitettiin keskelle otsan yläosaa. Otsan yläosa soveltuu tähän tarkoitukseen hyvin, koska alueella ei ole huomattavia kasvolihaksia. Elektrodien asettelu tapahtui Fridlundin ja Cacioppon [16] ohjeistuksen mukaan.

Elektrodien muodostamat signaalit vahvistettiin ja kaistanpäästösodatettiin Grass Model 15 Neurodata Acquisition System -järjestelmällä. Sekä *corrugator supercilii* että *zygomaticus major* -kanavien päästökaistat olivat 10–1000 Hz. Mittausvaiheessa käytettiin suhteellisen leveitä päästökaistoja, koska myöhemmin niitä voitaisiin kaventaa digitaalisella suodatuksella. Silmänräpäyskanavan päästökaista oli 0,3–30 Hz. *Corrugator supercilii* ja *zygomaticus major* -kanavien vahvistus oli 10 000 kertainen ja silmänräpäyskanavan 2000 kertainen.

Analogiset EMG-signaalit digitoitiin National Instruments AT-MIO-64E-3 -digitoitinkortilla, jonka amplitudiresoluutio on 12 bittiä. Signaalit näytteistettiin 3000 Hz:n näytteenottotaajuudella. Näytteenottoteoreeman kannalta 2000 Hz olisi tässä tapauksessa riittävän suuri näytteenottotaajuus. Vahvistimen suodatimet eivät ole kuitenkaan ideaalisia, jolloin on parempi käyttää hieman suurempaa näytteenottotaajuutta laskostumisen välttämiseksi. Päätökseen vaikutti myös Fridlundin ja Cacioppon [16] suositus näytteenottotaajuuksista: neljästä kahdeksaan kertaa niin suuri näytteenottotaajuus kuin signaalissa esiintyvä korkein taajuuskomponentti. Tämä olisi kuitenkin kasvattanut digitoitujen signaalien vaatimaa levytilaa, minkä tähden 3000 Hz:n näytteenottotaajuus koettiin sopivaksi. Mittausohjelmistona käytettiin Grass-Telefactor Gamma (versio 4.2) -ohjelmaa. Mittausasetelma on esitetty kaaviomuodossa kuvassa 5.4.

EMG-signaaleja mitattiin kolmeltakymmeneltä koehenkilöltä. Joukko koostui



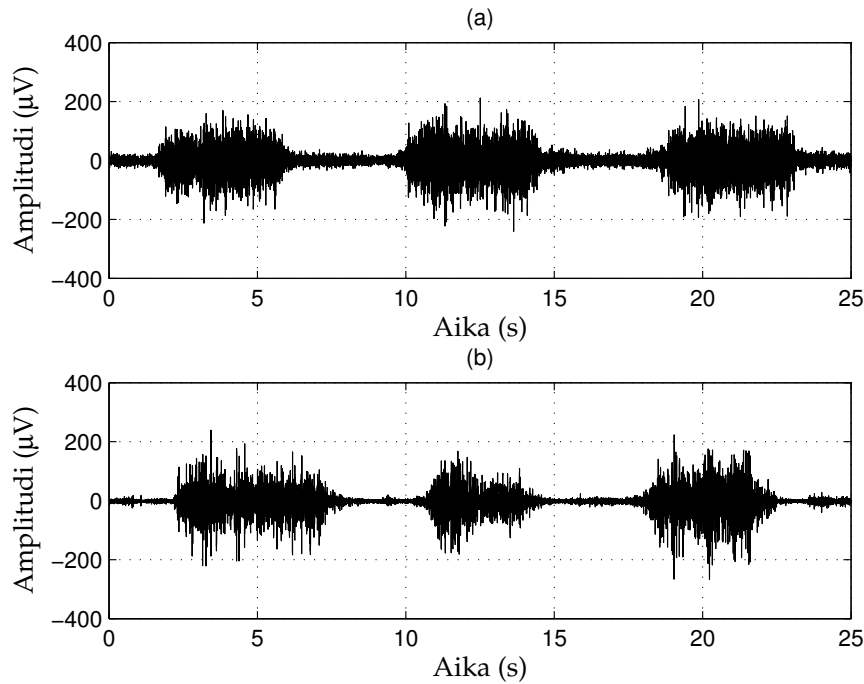
pääasiassa opiskelijoista, joiden keski-ikä oli 24,17 vuotta (keskihajonta: 3,53 vuotta). Nuorin koehenkilö oli 20-vuotias ja vanhin oli 35-vuotias. Suurin osa koehenkilöistä oli miehiä; naisia oli 8. Tutkielmassa ei kuitenkaan tarkastella iän, sukupuolen tai muiden vastaavien tekijöiden vaikutusta EMG-signaalien muotoon ja ominaisuuksiin. Lisäksi EMG-signaalien ominaisuuksissa esiintyy eroja iästä tai sukupuolesta riippumatta. Tämä selittyy pääasiassa siten, että ihmiset ovat erilaisia.

Jokaista koehenkilöä kohden varattiin 1,5 tuntia aikaa mittausten suorittamiseen. Itse mittaaminen kesti n. 40–60 minuuttia, mutta elektrodien kiinnittämisen, koehenkilön ohjeistuksen sekä lyhyiden taukojen johdosta varattiin hieman lisää aikaa. Aluksi koehenkilölle kerrottiin yleistä tietoa mittauksesta ja annettiin alustava ohjeistus toiminnasta mittauksen aikana. Sen jälkeen koehenkilön kasvot puhdistettiin mittaushaavien alueilta ja elektrodit kiinnitettiin vastaaville alueille.

Mittaaminen jakaantui neljään vaiheeseen. Ensimmäisessä mittausvaiheessa koehenkilö aktivoi *corrugator supercilii* -kasvolihasta otsaansa rypistämällä. Toisessa mittausvaiheessa koehenkilö aktivoi *zygomaticus major* -kasvolihasta hymyilemällä. Kolmannessa mittausvaiheessa koehenkilö aktivoi molempia lihaksia yhtä aikaa. Kaikissa mittausvaiheissa yksi aktivaatio kesti noin kahden sekunnin ajan, minkä jälkeen aktivoitu lihas annettiin olla rentona toiset kaksi sekuntia. Neljännessä mittausvaiheessa suoritettiin perustasomittaus, minkä aikana koehenkilön tuli pitää kasvot rentoina. Jokainen mittausvaihe kesti kymmenen minuuttia. Koehenkilöitä ohjeistettiin pitämään pidempiä taukoja aktivaatioiden välillä mikäli se olisi tarpeellista. Näin pyrittiin suorittamaan jokainen mittausvaihe kerralla loppuun asti. Koehenkilöillä oli myös mahdollisuus pitää pidempiä taukoja mittausvaiheiden välillä.

Aktivaatioiden ja taukojen pituudet eivät olleet täsmälleen kahden sekunnin mittaisia. Siitä huolimatta lihasten aktivaatioita saatiin mitattua riittävästi. Jos nimittäin yhden aktivaatio ja lepo -vaiheen keskimääräiseksi pituudeksi arvioidaan 5 s, niin silloin henkilöä kohden rekisteröitiin noin 120 lihasaktivaatiota. Kun aktivaatioita mitattiin kolmessa mittausvaiheessa, niin silloin henkilöä kohden on yhteensä noin 480 aktivaatiota. Tässä on otettu huomioon kolmannen vaiheen molemmat mittauskanavat.

Ihon impedanssi mitattiin elektrodien kiinnityskohdista ennen ensimmäistä mittausvaihetta, mittausvaiheiden välissä ja viimeisen mittausvaiheen jälkeen. Tällä tavalla voitiin seurata impedanssiarvojen kehittymistä ja varmistua siitä, että ne pysyivät keskimäärin 10 k $\Omega$ :n alapuolella koko mittauksen ajan. Jos impedanssiarvot olivat suurempia, kiinnitettiin kyseiset elektrodit uudelleen. Vastaavalla tavalla toimittiin elektrodien irrotaessa. Elektrodien irtoamisesta seurasi mittausvaiheen suorittaminen uudelleen. Näin ei kuitenkaan toimittu, jos mittausvaihetta



Kuva 5.5: Signaalinäytteet *corrugator supercilii* (a) ja *zygomaticus major* (b) -kasvolihasten alueelta mitatuista EMG-signaaleista. Signaaleissa on selvästi havaittavissa kolme erillistä lihasten aktivaatiota.

ei enää ollut paljoa jäljellä.

Kuvassa 5.5 on 25 sekunnin pituiset signaalinäytteet *corrugator supercilii* ja *zygomaticus major* -kasvolihasten alueilta mitatuista EMG-signaaleista. Kuvassa voidaan selvästi havaita kolme lihasten aktivaatiota, jotka esiintyvät signaalin amplitudiarvojen voimistumisena. Tutkielmassa käytetyn varsinaisen tutkimusaineiston muodostivat vain ensimmäisen mittausvaiheen *corrugator supercilii* -kasvolihaksen alueelta mitatut signaalit. Tämä ei haittaa, koska *corrugator supercilii* ja *zygomaticus major* -kasvolihasten signaaleissa ei ainakaan aktivaation mielessä näyttäisi esiintyvän kovinkaan suuria eroavuuksia.

Jokaisessa mittausvaiheessa tallennettiin kaikkien mittauskanavien signaalit. Myös koehenkilöiden tiedot (ikä ja sukupuoli) ja mittaustiedot (mittauspäivämäärä ja -aika, näytteenottotaajuus, vahvistimen kanavakohtaiset asetukset, impedanssiarvot mittausvaiheiden välillä sekä mittausvaiheiden aloitus- ja lopetusajat) kirjattiin ja arkistoitii. Tämän johdosta signaaleja on mahdollista hyödyntää tulevaisuudessa myös muissa tutkimuksissa. Tutkimusaineistoa kerättiin suhteellisen paljon ja sitä onkin riittävästi mahdollisten jatkotutkimusten materiaaliksi.

## 5.3 Testien toteutus

Testien tarkoituksena oli tarkastella MLP-neuroverkkojen toimintaa osana suunniteltua ilmaisinjärjestelmää. Erityisesti tarkasteltiin ilmaisinyksikön toimintaa, kun MLP-neuroverkon rakennetta varioitiin piiloneuronien osalta. Lisäksi kokeiltiin erilaisten kaistanpäästösuodattimien ja aalloke-kohinanpoiston käyttämistä esikäsittelyssä. Esikäsittelymenetelmien testaaminen ei olisi ollut järkevää mikäli sitä ei olisi suoritettu vaihtelevissa olosuhteissa. Sen tähden esikäsittelymenetelmien toimintaa kokeiltiin erilaisilla kohinatasoilla.

Testit toteutettiin Matlab (versio 7.0) -ohjelmistolla. Matlab soveltui tehtävään hyvin, koska siihen liitettävien lisäosien avulla käyttöön saatiin sopivia työkaluja. Tässä tapauksessa käytettiin Neural Network Toolbox (versio 4.0.3), Signal Processing Toolbox (versio 6.2), Statistics Toolbox (versio 5.0) ja Wavelet Toolbox (versio 3.0) -lisäosia. Matlabin soveltuvuutta lisäsi se, ettei tarkoituksena vielä tässä vaiheessa ollut valmiin reaaliaikaisen järjestelmän toteuttaminen. Yksityiskohtaisemmat kuvaukset ohjelmistosta ja sen toiminnoista löytyy The Mathworksin kotisivuilta (<http://www.mathworks.com>).

Yhteensä testattiin seitsemää MLP-neuroverkkoa. Jokaisessa neuroverkossa oli kaksi syöteneuronia ja kolme tulosneuronia. Aktivaatiofunktioina käytettiin yksinään logistisia funktioita. Neuroverkot erosivatkin toisistaan ainoastaan piilokerrosten ja piiloneuronien lukumäärien suhteen. Pienimmässä neuroverkossa ei ollut lainkaan piiloneuroneja (2–0–3 -verkko) ja se simuloikin yksikerroksista perceptroonia. Viisi neuroverkosta oli kaksikerroksisia ja niissä oli yhdestä viiteen piiloneuronia yhdessä piilokerroksessa (kutsutaan 2– $h$ –3 -verkoiksi, missä  $h$  on piiloneuronien lukumäärä). Suurin kaksikerroksinen neuroverkko noudattaa Kolmogorovin teoreemaa, koska siinä on 5 piiloneuronia. Edellisten lisäksi kokeiltiin kolmikerroksista MLP-neuroverkkoa, jonka molemmissa piilokerroksissa oli viisi piiloneuronia (2–5–5–3 -verkko).

Neuroverkkoja opetettiin 1024:n näytteen pituisten signaali-ikkunoiden perusteella muodostettujen syötevektorien ja niihin liittyvien tulosvektorien avulla. Syötevektorit koostuivat kahdesta arvosta, jotka muodostettiin aikaisemmin tässä luvussa esitellyllä RMS-arvoihin perustuvalla menetelmällä. Tulosvektorit olivat binaariarvoisia vektoreita, joissa ikkunan luokkaa vastaava elementti oli arvoltaan yksi ja muut elementit olivat nollia.

Signaali-ikkunan pituutta valittaessa ajateltiin aalloke-kohinanpoistoa, jossa pyramidalgoritmi edellyttää signaalin pituuden olevan luvun kaksi potenssi. Koska käytettiin aitoja EMG-signaaleja, eikä keinotekoisien mallien perusteella generoituja signaaleja, oli huomattavasti vaikeampaa määrittää tarkasti motoristen tapahtumien esiintymishetkiä. Leveämmän signaali-ikkunan käyttäminen helpotti motoristen tapahtumien etsimistä, koska tarkkoja esiintymishetkiä ei tarvinnut määrittää. Riittävä tarkkuus saavutettiin etsimällä signaali-ikkunan pituisia segmenttejä,

joiden alueilla motoriset tapahtumat esiintyivät.

Signaali-ikkunoiden luokittelu vastasi aikaisemmin esiteltyä kolmiluokkaista luokittelua: lihasaktiviteetin alkaminen, tasannekohta ja lihasaktiviteetin päättyminen. Ensimmäinen ja viimeinen luokka vastaavat motorisia tapahtumia ja keskimäinen luokka muita tapauksia. Signaali-ikkunoiden valinta tapahtui manuaalisesti. Jokaisen koehenkilön signaali käytiin läpi ja siitä valittiin 48 signaali-ikkunaa (16 signaali-ikkunaa yhtä luokkaa kohden). Signaali-ikkunat pyrittiin valitsemaan tasaisesti koko signaalin alueelta, koska niiden haluttiin esittävän vaihtelevasti lihasaktiviteetin eri vaiheita. Tasannekohtia vastaavista ikkunoista puolet sisälsivät päällä olevaa lihasaktiviteettia.

Neuroverkkojen opettamisessa ja testaamisessa käytetty datajoukko on usein tapana jakaa *opetus-, validointi- ja testijoukkoihin* (training, validation and test sets). Validointijoukon avulla neuroverkon suorituskykyä voidaan seurata oppimisprosessin aikana. Käytännössä verkkoa simuloidaan validointijoukolla aina sen jälkeen, kun painokertoimet ovat muuttuneet. Tavoitteena on neuroverkon ylioppimisen välttäminen, josta seuraa yleistämiskyvyn heikkeneminen. Testijoukkoa käytetään oppineen neuroverkon testaamiseen. Joukkojen tulisi olla sellaisia, ettei niissä ole samoja alkioita. Silloin neuroverkon oppimista voidaan seurata luotettavasti ja testitulokset eivät ole liian positiivisia.

Kolmekymmentä koehenkilöä jaettiin satunnaisesti opetus-, validointi- ja testijoukkoihin. Opetusjoukko koostui 14:stä, validointijoukko 4:stä ja testijoukkoa 12:sta koehenkilöstä. Aikaisemmin todettiin, että jokaista koehenkilöä kohden valittiin 48 signaali-ikkunaa. Tämän perusteella opetusjoukossa oli 672, validointijoukossa 192 ja testijoukossa 576 alkioita. Yksikään alkioista ei kuulunut useampaan kuin yhteen joukkoon. Riittävän yleistämiskyvyn saavuttamiseksi opetusjoukon koon  $N$  tulisi täyttää ehto

$$N \geq W/\epsilon,$$

missä  $W$  on neuroverkon painokertoimien lukumäärä ja  $\epsilon$  on virheellisten luokittelujen osuus [21][41]. Kun toivotuksi virheiden osuudeksi asetetaan 10 % ja lasketaan kertoimien lukumäärä suurimman neuroverkon (kolmikerroksisen 2–5–5–3-verkon) mukaan, niin opetusjoukon kooksi saadaan 630 alkioita. Tässä on huomioitu myös bias-painojen lukumäärä. Jos tulosta verrataan opetusjoukon kokoon, huomataan opetusjoukon olevan suurempi. Tämä johtuu siitä, että tutkimuksen alkuvaiheessa pohdittiin myös suurempien neuroverkkojen käyttämistä. Ikkunoita valittiin enemmän, ettei opetusjoukon pienestä koosta olisi seurannut jatkossa ongelmia.

Neuroverkkojen opettamiseen käytettiin laskeutuvan gradientin menetelmän versiota, jossa sovelletaan mukautuvaa oppimisnopeutta ja momenttitermiä. Neural Network Toolboxissa vastaavan algoritmin nimi on `traindx`. Kyseinen algoritmi on batch-tyyppinen, eli kaikki opetusjoukon alkiot käydään läpi ennen neurover-

Taulukko 5.1: Oppimisalgoritmin parametrit.

Parametri	Arvo	Kuvaus
epoch	1000	Iteraatioiden (epookkien) lukumäärä.
goal	0	Virhefunktion tavoitearvo.
lr	0,01	Oppimisnopeus.
lr_inc	1,05	Oppimisnopeuden nopeutuskerroin.
lr_dec	0,7	Oppimisnopeuden hidastuskerroin.
max_fail	5	Validointivirheiden suurin sallittu lukumäärä.
max_perf_inc	1,04	Suurin sallittu virhefunktion muutosnopeus.
mc	0,9	Momenttitermi.
min_grad	$1 \cdot 10^{-6}$	Pienin sallittu gradientin muutosnopeus.
show	25	Välitulosten esitystaaajuus.
time	inf	Opetuksen aikarajoitus.

kon painokertoimien muuttamista. Toisin sanoen painokertoimien muutos vastaa opetusjoukon kaikkien alkioden mukaista keskimääräistä muutosta. Opetuksen aikana käytettiin pääasiassa `traingdx:n` vakioparametreja. Ainoastaan iteraatioiden lukumäärä asetettiin vakioarvosta poikkeavaksi. Taulukossa 5.1 on esitetty testeissä käytetyt parametrien arvot.

Mukautuvaa oppimisnopeutta soveltavaa algoritmia käytettiin, koska optimaalisen oppimisnopeuden määrittäminen olisi edellyttänyt useiden testiajojen suorittamista. Kiinteä oppimisnopeus ei välttämättä olisi edes toiminut hyvin opetuksen edetessä. Parempi vaihtoehto olikin antaa oppimisnopeuden mukautua sopivaksi opetuksen aikana. Momenttitermin käytöllä painokertoimiin tehtäviä muutoksia pyritään tasoittamaan siten, että muutokset ohjautuisivat optimaalisempaan suuntaan.

Neuroverkon virhefunktiona käytettiin *keskineliövirhettä* (mean squared error). Se määritellään kaavalla

$$MSE = \frac{1}{N} \sum_{n=0}^{N-1} (y_n - t_n)^2,$$

missä  $y_n$  on neuroverkon tulos  $n$ :lle syötteelle,  $t_n$  on syötettä vastaava tavoitetulos ja  $N$  on alkioden lukumäärä. Virhefunktion arvo lasketaan jokaisella oppimisalgoritmin iteraatiolla. Neuroverkon opettaminen pysäytettiin, jos validointijoukon keskineliövirhe kasvoi huomattavasti. Tällä tavalla pyrittiin välttämään neuroverkon ylioppiminen.

Neuroverkkojen painokertoimien alkuarvot valittiin Russon säännön [29] perusteella. Sen mukaan neuroverkon painokertoimet valitaan satunnaisesti suljetulta reaali lukuväliltä  $\left[-\frac{2,4}{M}, \frac{2,4}{M}\right]$ , missä  $M$  on neuroniin saapuvien yhteyksien lu-

Taulukko 5.2: Aaloke-kohinanpoiston parametrit.

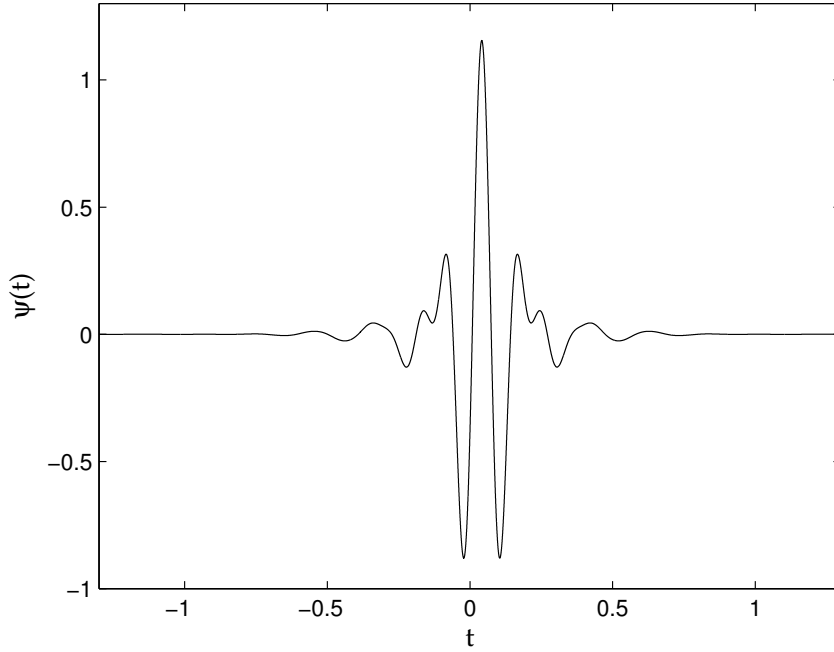
Parametri	Arvo	Kuvaus
<code>tptr</code>	<code>rigsure</code>	Kynnystyyppi.
<code>sorh</code>	<code>s</code>	Kynnystystapa.
<code>scal</code>	<code>sln</code>	Kynnysarvojen skaalaus.
<code>n</code>	<code>3</code>	Aaloke-muunnoksen syvyys.
<code>wname</code>	<code>dmey</code>	Äitiaaloke.

kumäärä. Tällä tavalla valittujen painokertoimien pitäisi estää oppimisnopeuden ennenaikainen hidastuminen eli *saturaatio* (saturation).

Neuroverkkojen toimintaa testattiin erilaisten esikäsittelymenetelmien tapauksessa. Niitä olivat 10–1000, 10–500 ja 10–250 Hz:n kaistanpäästösuodattimet ja aaloke-kohinanpoisto. Kaistanpäästösuodattimien vaimennus estokaistoilla oli 40 desibeliä. Kaistanpäästösuodatuksessa sekä suodatettava signaali-ikkuna että suodatin täydennettiin nolilla sopivan pituisiksi, jotta suodatuksen tulos vastaisi lineaarista konvoluutiota. Suodatukseseen käytettiin Signal Processing Toolboxin `fftfilt`-funktioita, joka perustuu FFT:n avulla toteutettuun nopeaan konvoluutioon (overlap-add). Suodatuksen tuloksena saadun signaali-ikkunan alusta ja lopusta poistettiin suodatuksesta syntyneet hännät (noin puolet suotimen pituudesta). Sen jälkeen tuloksena saatiin 1024 näytteen pituinen suodatettu signaali-ikkuna.

Aaloke-kohinanpoistoon käytettiin Wavelet Toolboxin `wden`-funktioita. Tässä yhteydessä käytettiin SURE-sääntöä kynnyn valinnassa, pehmeää kynnystystä, ensimmäisen detaljitason perusteella lasketun kohinatasoestimaatin mukaista kynnysarvojen skaalausta, Meyer-aalokkeen FIR-pohjaista approksimaatiota (kuva 5.6) sekä kolmannelle tasolle ulottuvaa aaloke-muunnosta. Meyer-aalokkeeseen päädyttiin, koska siihen liittyvät aaloke-suodattimet leikkasivat taajuustason jyrkimmin muihin Matlabin aaloke-suodattimiin verrattuna. Tämän seurauksena muunnoksen eri tasot käsittävät vähemmän ylempien tasojen informaatiota. Toisin sanoen eri detaljitason erottelevat paremmin signaalin taajuusinformaatiota. Kolmannelle tasolle asti tehty muunnos osoittautui alustavissa testeissä riittävän, eikä muunnosta sen tähden tehty syvemmille tasoille. Taulukossa 5.2 on yhteenveto `wden`-funktion parametrien arvoista.

Opetettujen neuroverkkojen testauksessa käytettiin kolmea testijoukkoa. Ensimmäinen koostui testijoukon alkioista sellaisenaan. Toisen testijoukon alkioihin lisättiin valkoista kohinaa sen verran, että signaali-ikkunoiden *signaalikohinasuhteeksi* (signal-to-noise ratio) saatiin 5 desibeliä. Kolmannen testijoukon alkioihin lisättiin myös valkoista kohinaa, mutta nyt signaalikohinasuhteeksi määriteltiin 1 desibeli. Jos valkoinen kohina määritellään *additiiviseksi* (additive), voidaan kon-



Kuva 5.6: Meyer-aallokkeen  $\Psi(t)$  kuvaaja.

taminoituneen signaalin  $y(n)$  malli kuvata yhtälöllä

$$y(n) = x(n) + az(n), \quad (5.2)$$

missä  $x(n)$  on alkuperäinen signaali,  $z(n)$  on kohinasignaali ja  $a$  on sopivasti valittu kerroin. Kerroin määrittää kohinasignaalin voimakkuuden. Tietyn signaalikohinasuhteen saavuttamiseksi onkin määritettävä sopiva arvo kertoimelle  $a$ . Mielivaltaisen signaalin  $x(n)$ , jonka pituus on  $N$ , teho määritellään kaavalla

$$P[x(n)] = \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2.$$

Nyt signaalin  $az(n)$  teho voidaan määritellä kaavalla

$$P[az(n)] = \frac{1}{N} \sum_{n=0}^{N-1} |az(n)|^2 = \frac{1}{N} \sum_{n=0}^{N-1} a^2 |z(n)|^2 = a^2 \frac{1}{N} \sum_{n=0}^{N-1} |z(n)|^2 = a^2 P[z(n)]. \quad (5.3)$$

Kaavan (5.2) signaalin  $y(n)$  signaalikohinasuhde määritellään kaavalla

$$SNR = 10 \log_{10} \frac{P[x(n)]}{P[az(n)]},$$

josta saadaan kaavaa (5.3) soveltamalla

$$a = \sqrt{\frac{P[x(n)]/P[z(n)]}{10^{SNR/10}}}.$$

Kohinaisten testijoukkojen muodostuksessa toimittiin seuraavasti. Signaalin tehoa arvioitiin jokaisen koehenkilön kohdalla erikseen. Arviona käytettiin koehenkilön signaali-ikkunoiden tehojen keskiarvoa. Näin voitiin varmistua siitä, että jokaiseen signaali-ikkunaan lisättiin suhteellisesti yhtä paljon valkoista kohinaa. Kohinasignaalin kohdalla teho laskettiin satunnaisesti generoidun kohinasignaalin perusteella.

Kun kohina lisätään edellä kuvatulla tavalla, niin tuloksena saatavan signaalin signaalikohinasuhde ei ole täsmälleen 5 tai 1 desibeliä. Tämä johtuu siitä, että alkuperäisissä signaaleissa on jo valmiiksi mittausvaiheessa ja A/D-muunnoksessa syntynyttä kohinaa. Signaalien signaalikohinasuhde on siis jonkin verran määriteltä huonompi. Myös signaalin tehon arviointiin sovellettu menetelmä vaikuttaa tulokseen. Edelliset seikat eivät haittaa, koska kohinan lisäämisen tarkoituksena oli vain havaitsemistehtävän vaikeuttaminen.

Opetus-, validointi- ja testijoukkojen alkiot normalisoitiin ennen neuroverkkojen opettamista ja testaamista. Tarkalleen ottaen normalisointi kohdistui syötevektoreihin. Tässä sovellettiin menetelmää, jota tilastotieteessä kutsutaan *standardoinniksi* (standardizing). Ensin laskettiin opetusjoukon syötevektoreiden perusteella neuroverkon jokaisen syötearvon keskiarvo ja keskihajonta. Sen jälkeen syötevektoreiden syötearvoista vähennettiin vastaavat keskiarvot ja tulokset jaettiin vielä vastaavilla keskihajonnoilla. Normalisoinnin tarkoituksena oli varmistaa, että kaikki syötearvot vaikuttavat yhtä voimakkaasti neuroverkon toimintaan. Edellinen toistettiin myös validointi- ja testijoukoille käyttäen opetusjoukon perusteella laskettuja tunnuslukuja. Normalisoinnissa käytettiin Neural Network Toolboxin `prestd` ja `trastd` -funktioita.

Opettaminen ja testaaminen toistettiin jokaiselle neuroverkko- ja esikäsitteilymenetelmäyhdistelmällä 100 kertaa. Koska opetuksessa käytettiin satunnaisia painokertoimien alkuarvoja, saatiin useampien toistojen perusteella parempi kuva siitä, kuinka ilmaisinyksikkö keskimäärin toimii. Neuroverkkojen opettaminen ja testaus toteutettiin Neural Network Toolboxin `train` ja `sim` -funktioilla.

## 5.4 Tulosten analysointia

Seuraavaksi analysoidaan aikaisemmin kuvattujen testien tuloksia. Tulokset esitetään yksityiskohtaisesti taulukoissa 5.3–5.6. Ensimmäisessä sarakkeessa on neuroverkon arkkitehtuuri (MLP) ja toisessa sarakkeessa on sen opettamiseen kuluneiden iteraatioiden lukumäärä ( $E$ ). Kolmannessa ja neljännessä sarakkeessa ovat



Taulukko 5.3: Testituloksia 10–1000 Hz:n kaistanpäästösuodatuksella.

MLP	$E$	$MSE_t$	$MSE_v$	$A_1$	$A_2$	$A_3$
2–0–3	321,3 (5,6)	0,0080 (0,0001)	0,0072 (0,0004)	95,84 % (0,40 %)	88,73 % (0,48 %)	74,57 % (0,85 %)
2–1–3	245,3 (13,7)	0,0834 (0,0145)	0,0835 (0,0146)	94,42 % (9,45 %)	93,68 % (8,73 %)	84,00 % (7,28 %)
2–2–3	263,3 (2,5)	0,0073 (0,0000)	0,0067 (0,0000)	96,35 % (0,00 %)	89,41 % (0,00 %)	76,44 % (0,13 %)
2–3–3	260,7 (2,4)	0,0074 (0,0000)	0,0067 (0,0000)	96,41 % (0,08 %)	89,41 % (0,00 %)	76,66 % (0,13 %)
2–4–3	257,2 (2,7)	0,0075 (0,0000)	0,0067 (0,0000)	96,48 % (0,08 %)	89,41 % (0,00 %)	76,73 % (0,04 %)
2–5–3	256,6 (2,1)	0,0075 (0,0000)	0,0067 (0,0000)	96,48 % (0,08 %)	89,41 % (0,00 %)	76,73 % (0,04 %)
2–5–5–3	236,9 (5,3)	0,0074 (0,0000)	0,0067 (0,0000)	96,37 % (0,07 %)	89,41 % (0,04 %)	76,59 % (0,21 %)

MLP on neuroverkon arkkitehtuuri,  $E$  on iteraatioiden lukumäärä,  $MSE_t$  on opetusjoukon keskineliövirhe,  $MSE_v$  on validointijoukon keskineliövirhe sekä  $A_1$ ,  $A_2$  ja  $A_3$  ovat kolmen testijoukon luokittelutarkkuudet. Arvot ovat sadan opetus-/testikerran keskiarvoja. Keskihajonnat ovat sulkeissa.

opetusjoukon ja validointijoukon keskineliövirheet ( $MSE_t$  ja  $MSE_v$ ). Kolmessa viimeisessä sarakkeessa ovat luokittelutarkkuudet ( $A_1$ ,  $A_2$  ja  $A_3$ ), jotka vastaavat sivulla 51 kuvattuja testijoukkoja. Luokittelutarkkuudet on laskettu jakamalla oikeiden luokittelujen lukumäärä testijoukon koolla. Ne kertovat kuinka monta prosenttia testijoukkojen tapauksista kyseinen verkko luokitteli oikein. Taulukoiden arvot ovat sadan opetus-/testikerran perusteella laskettuja keskiarvoja. Keskihajonnat on esitetty sulkeissa.

Taulukossa 5.3 on 10–1000 Hz:n kaistanpäästösuodatuksella saatuja testituloksia. Lähes kaikkien neuroverkkojen opettaminen onnistui alle kolmellasadalla oppimisalgoritmin iteraatiolla. Ainoastaan yksikerroksisen neuroverkon kohdalla tarvittiin yli kolmesataa iteraatiota. Keskineliövirheiden perusteella eri kokoisilla neuroverkoilla saavutettiin yhtä hyvät painokertoimien arvot. Myös luokittelutarkkuudet tukevat tätä johtopäätöstä. Ainoastaan 2–1–3 -verkolla saavutetaan hieman huonommat keskineliövirheet. Tämä ei kuitenkaan haittaa merkittävästi, koska ensimmäisen testijoukon kohdalla 2–1–3 -verkko on vain pari prosenttiyksikköä muita neuroverkkoja heikompi. Kohinaisten testijoukkojen kohdalla sen avulla saavutetaan parhaimmat luokittelutarkkuudet. Kohinan vaikutus havaitaan selvästi, kun tarkastellaan testijoukkojen luokittelutarkkuuksien välisiä eroja. Ensimmäisen ja kolmannen testijoukon välillä esiintyy jopa 20 prosenttiyksikön suuruisia eroja. Ne

Taulukko 5.4: Testituloksia 10–500 Hz:n kaistanpäästösuodatuksella.

MLP	$E$	$MSE_t$	$MSE_v$	$A_1$	$A_2$	$A_3$
2–0–3	321,3 (6,5)	0,0084 (0,0001)	0,0069 (0,0003)	95,63 % (0,23 %)	93,01 % (0,36 %)	86,48 % (0,81 %)
2–1–3	243,3 (18,1)	0,0843 (0,0199)	0,0845 (0,0200)	94,39 % (8,77 %)	95,19 % (8,89 %)	91,60 % (8,46 %)
2–2–3	272,4 (4,7)	0,0079 (0,0001)	0,0063 (0,0001)	95,58 % (0,09 %)	93,21 % (0,12 %)	86,62 % (0,24 %)
2–3–3	269,3 (4,7)	0,0080 (0,0001)	0,0063 (0,0000)	95,68 % (0,11 %)	93,23 % (0,08 %)	86,78 % (0,26 %)
2–4–3	266,5 (3,7)	0,0080 (0,0000)	0,0063 (0,0000)	95,83 % (0,12 %)	93,28 % (0,08 %)	87,04 % (0,14 %)
2–5–3	265,6 (4,0)	0,0080 (0,0000)	0,0063 (0,0000)	95,90 % (0,12 %)	93,34 % (0,08 %)	87,10 % (0,10 %)
2–5–5–3	238,9 (21,5)	0,0123 (0,0304)	0,0107 (0,0306)	94,16 % (10,65 %)	91,79 % (9,69 %)	85,56 % (8,12 %)

MLP on neuroverkon arkkitehtuuri,  $E$  on iteraatioiden lukumäärä,  $MSE_t$  on opetusjoukon keskineliövirhe,  $MSE_v$  on validointijoukon keskineliövirhe sekä  $A_1$ ,  $A_2$  ja  $A_3$  ovat kolmen testijoukon luokittelutarkkuudet. Arvot ovat sadan opetus-/testikerran keskiarvoja. Keskihajonnat ovat sulkeissa.

selittyvät siten, että kohina pääsee vaikuttamaan leveän päästökaistan johdosta useilla taajuuksilla.

Taulukossa 5.4 on 10–500 Hz:n kaistanpäästösuodatuksella saatuja testituloksia. Päästökaistan kaventumisella ei ole merkittävää vaikutusta iteraatioiden lukumäärään. Myös keskineliövirheet ovat vastaavanlaisia kuin edellä. Suurimmat erot 10–1000 Hz:n kaistanpäästösuodattamiseen verrattuna esiintyvätkin luokittelutarkkuuksissa. Ne parantuvat kohinaisten testijoukkojen kohdalla. Kyseessä on huomattavasti kapeampi päästökaista, jolloin suurempi osa kohinasta suodattuu pois. Samalla aikaa EMG-signaalin merkittävät taajuuskomponentit vaikuttavat voimakkaammin. Tällä kertaa 2–1–3 -verkko erottuu selvästi muista neuroverkoista vain kohinaisimman testijoukon kohdalla.

Taulukossa 5.5 on 10–250 Hz:n kaistanpäästösuodatuksella saatuja testituloksia. Kyseessä on kapein kolmesta kaistanpäästösuodattimesta. Aikaisempiin tuloksiin verrattuna suurimmat erot esiintyvät luokittelutarkkuuksissa. Lievää keskineliövirheiden huonontumista on tosin havaittavissa, mutta se ei vaikuta negatiivisesti luokittelutarkkuuksiin. Testijoukkojen luokittelutarkkuuksien väliset erot ovat pienempiä kuin leveämpien päästökaistojen tapauksessa. Lisäksi 2–1–3 -verkko ei enää erotu muista neuroverkoista juuri lainkaan. Ainoastaan kohinaisimman testijoukon tapauksessa sen avulla saavutetaan paras tulos. Silloinkin erot ovat vain

Taulukko 5.5: Testituloksia 10–250 Hz:n kaistanpäästösudatuksella.

MLP	$E$	$MSE_t$	$MSE_v$	$A_1$	$A_2$	$A_3$
2–0–3	324,0 (7,3)	0,0093 (0,0002)	0,0087 (0,0007)	95,71 % (0,15 %)	95,41 % (0,39 %)	91,64 % (0,27 %)
2–1–3	242,8 (17,7)	0,0849 (0,0197)	0,0861 (0,0196)	94,25 % (9,52 %)	95,00 % (9,39 %)	94,07 % (8,92 %)
2–2–3	265,5 (5,9)	0,0090 (0,0000)	0,0080 (0,0001)	96,01 % (0,03 %)	95,83 % (0,00 %)	92,01 % (0,04 %)
2–3–3	265,2 (7,3)	0,0090 (0,0000)	0,0081 (0,0001)	96,09 % (0,09 %)	95,83 % (0,00 %)	92,11 % (0,12 %)
2–4–3	264,4 (5,1)	0,0090 (0,0000)	0,0081 (0,0002)	96,16 % (0,07 %)	95,83 % (0,00 %)	92,24 % (0,13 %)
2–5–3	263,9 (6,5)	0,0090 (0,0000)	0,0082 (0,0002)	96,18 % (0,09 %)	95,83 % (0,07 %)	92,31 % (0,12 %)
2–5–5–3	231,9 (15,9)	0,0111 (0,0212)	0,0102 (0,0213)	95,46 % (5,80 %)	95,21 % (5,94 %)	91,48 % (5,67 %)

MLP on neuroverkon arkkitehtuuri,  $E$  on iteraatioiden lukumäärä,  $MSE_t$  on opetusjoukon keskineliövirhe,  $MSE_v$  on validointijoukon keskineliövirhe sekä  $A_1$ ,  $A_2$  ja  $A_3$  ovat kolmen testijoukon luokittelutarkkuudet. Arvot ovat sadan opetus-/testikerran keskiarvoja. Keskihajonnat ovat sulkeissa.

parin prosenttiyksikön luokkaa.

Taulukossa 5.6 on aalloke-kohinanpoistolla saatuja testituloksia. Kyseisellä esikäsittelymenetelmällä kaikkien neuroverkkojen opettamiseen tarvittiin selvästi alle kolmesataa iteraatiota. Myös validointijoukon keskineliövirheissä voidaan havaita muutoksia, mutta tällä kertaa huonompaan suuntaan. Luokittelutarkkuuksissa on havaittavissa parantumista ja ylipäänsä aalloke-kohinanpoistolla saavutetaan parhaimmat luokittelutarkkuudet Kun aalloke-kohinanpoistolla saatuja luokittelutarkkuuksia vertaillaan keskenään, niin toisen testijoukon kohdalla on havaittavissa parempia luokittelutarkkuuksia ensimmäiseen testijoukkoon verrattuna. Ilmiö esiintyy 2–1–3 -verkolla myös 10–500 Hz:n ja 10–250 Hz:n kaistanpäästösudatusten tapauksessa. Ilmiölle löytyy selitys, kun tutkitaan testijoukkojen luokkohtaisia luokittelutarkkuuksia.

Luokkakohtaiset luokittelutarkkuudet esitetään taulukoissa 5.7–5.10. Ensimmäisessä sarakkeessa on neuroverkon arkkitehtuuri edellisten taulukoiden tavoin. Luokkakohtaiset luokittelutarkkuudet ( $C_1$ ,  $C_2$  ja  $C_3$ ) esitetään kolmen ryhmässä. Ensimmäinen luokka vastaa lihasaktiviteetin alkamista, toinen luokka tasannekoh-tia ja kolmas luokka lihasaktiviteetin päättymistä. Jokainen näistä kolmen luokan ryhmästä vastaa edelleen tiettyä testijoukkoa ( $A_1$ ,  $A_2$  ja  $A_3$ ). Luokkakohtaiset luokittelutarkkuudet on laskettu jakamalla yksittäisen luokan oikeiden luokittelujen

Taulukko 5.6: Testituloksia aalloke-kohinanpoistolla.

MLP	$E$	$MSE_t$	$MSE_v$	$A_1$	$A_2$	$A_3$
2-0-3	278,0 (0,0)	0,0092 (0,0000)	0,0099 (0,0000)	96,53 % (0,00 %)	96,35 % (0,00 %)	93,23 % (0,00 %)
2-1-3	245,6 (13,9)	0,0832 (0,0141)	0,0846 (0,0141)	94,79 % (6,21 %)	95,30 % (6,27 %)	94,54 % (6,21 %)
2-2-3	222,9 (3,9)	0,0086 (0,0001)	0,0109 (0,0002)	96,49 % (0,08 %)	96,93 % (0,14 %)	93,37 % (0,08 %)
2-3-3	220,2 (2,7)	0,0086 (0,0001)	0,0106 (0,0002)	96,50 % (0,07 %)	97,01 % (0,10 %)	93,37 % (0,07 %)
2-4-3	215,8 (5,1)	0,0087 (0,0001)	0,0104 (0,0001)	96,50 % (0,07 %)	97,06 % (0,12 %)	93,37 % (0,07 %)
2-5-3	214,1 (4,3)	0,0087 (0,0001)	0,0103 (0,0001)	96,51 % (0,06 %)	97,09 % (0,14 %)	93,38 % (0,06 %)
2-5-5-3	206,3 (3,7)	0,0085 (0,0001)	0,0108 (0,0001)	96,50 % (0,07 %)	97,03 % (0,18 %)	93,50 % (0,20 %)

MLP on neuroverkon arkkitehtuuri,  $E$  on iteraatioiden lukumäärä,  $MSE_t$  on opetusjoukon keskineliövirhe,  $MSE_v$  on validointijoukon keskineliövirhe sekä  $A_1$ ,  $A_2$  ja  $A_3$  ovat kolmen testijoukon luokittelutarkkuudet. Arvot ovat sadan opetus-/testikerran keskiarvoja. Keskihajonnat ovat sulkeissa.

lukumäärä kyseisen luokan koolla. Näin saatu suhdeluku kertoo kuinka monta prosenttia yksittäisen luokan tapauksista on luokiteltu oikein. Taulukoiden arvot ovat keskiarvoja, jotka on laskettu sadan opetus-/testikerran perusteella. Keskihajonnat on esitetty sulkeissa.

Luokkakohtaiset luokittelutarkkuudet noudattavat pääasiassa edellä esitettyjä kokonaistarkkuuksia. Luokkakohtainen tulosten tutkiminen kuitenkin paljastaa, että valkoisen kohinan lisääminen parantaa toisen luokan (tasannekohdat) luokittelutarkkuutta. Se selittää edellä havaitun ilmiön, jossa kohinan lisääminen paransi kokonaistarkkuuksia. Toisen luokan luokittelutarkkuuden parantuminen johtuu todennäköisesti siitä, että kohinan lisääminen tasoittaa sopivasti vaikeampia signaali-ikkunoita. Näin ollen epäselvemmät toisen luokan signaali-ikkunat luokituvat hieman paremmin. On kuitenkin syytä huomioida, ettei kokonaistarkkuuden parantumisesta seuraa, että myös muiden luokkien kohdalla kävisi vastaavalla tavalla. Ensimmäisen ja kolmannen luokan luokittelutarkkuuksissa onkin havaittavissa heikkenemistä. Toisen luokan parantuminen kuitenkin kompensoi sen verran, että kokonaistarkkuus parantuu. Luokkakohtaisista luokittelutarkkuuksista voidaan havaita myös se, että kolmas luokka oli useimmissa tilanteissa vaikein luokitella. 2-1-3 -verkon kohdalla on toisinaan kyllä niinkin, että toisen luokan kohdalla on heikoin luokittelutarkkuus.

Taulukko 5.7: Luokkakohtaiset luokittelutarkkuudet 10–1000 Hz:n kaistanpäästösuodatuksella.

MLP	$A_1$			$A_2$			$A_3$		
	$C_1$	$C_2$	$C_3$	$C_1$	$C_2$	$C_3$	$C_1$	$C_2$	$C_3$
2-0-3	98,44 % (0,00 %)	96,88 % (0,00 %)	92,20 % (1,19 %)	95,72 % (0,21 %)	99,48 % (0,00 %)	70,99 % (1,28 %)	85,80 % (0,56 %)	100,0 % (0,00 %)	37,90 % (2,12 %)
2-1-3	98,25 % (9,95 %)	90,09 % (9,12 %)	94,92 % (10,90 %)	96,26 % (10,04 %)	97,58 % (9,81 %)	87,21 % (8,07 %)	92,83 % (11,43 %)	98,06 % (9,91 %)	61,09 % (7,64 %)
2-2-3	98,44 % (0,00 %)	96,35 % (0,00 %)	94,27 % (0,00 %)	95,83 % (0,00 %)	99,48 % (0,00 %)	72,92 % (0,00 %)	86,98 % (0,00 %)	100,0 % (0,00 %)	42,33 % (0,39 %)
2-3-3	98,44 % (0,00 %)	96,51 % (0,24 %)	94,27 % (0,00 %)	95,83 % (0,00 %)	99,48 % (0,00 %)	72,92 % (0,00 %)	86,98 % (0,00 %)	100,0 % (0,00 %)	43,01 % (0,40 %)
2-4-3	98,44 % (0,00 %)	96,72 % (0,24 %)	94,27 % (0,00 %)	95,83 % (0,00 %)	99,48 % (0,00 %)	72,92 % (0,00 %)	86,98 % (0,00 %)	100,0 % (0,00 %)	43,21 % (0,13 %)
2-5-3	98,44 % (0,00 %)	96,73 % (0,23 %)	94,27 % (0,00 %)	95,83 % (0,00 %)	99,48 % (0,00 %)	72,92 % (0,00 %)	86,97 % (0,05 %)	100,0 % (0,00 %)	43,22 % (0,10 %)
2-5-5-3	98,44 % (0,00 %)	96,42 % (0,19 %)	94,27 % (0,05 %)	95,83 % (0,00 %)	99,48 % (0,00 %)	72,91 % (0,10 %)	87,00 % (0,10 %)	100,0 % (0,00 %)	42,76 % (0,64 %)

MLP on neuroverkon arkkitehtuuri. Luokkakohtaiset luokittelutarkkuudet  $C_1$ ,  $C_2$  ja  $C_3$  on ryhmitelty kolmen testijoukon  $A_1$ ,  $A_2$  ja  $A_3$  mukaan. Arvot ovat sadan opetus-/testikerran keskiarvoja. Keskihajonnat ovat sulkeissa.

Taulukko 5.8: Luokakokohtaiset luokittelutarkkuudet 10–500 Hz:n kaistanpäästösuodatuksella.

MLP	$A_1$			$A_2$			$A_3$		
	$C_1$	$C_2$	$C_3$	$C_1$	$C_2$	$C_3$	$C_1$	$C_2$	$C_3$
2-0-3	98,44 % (0,00 %)	96,88 % (0,00 %)	91,56 % (0,69 %)	97,70 % (0,26 %)	99,48 % (0,00 %)	81,83 % (0,83 %)	94,19 % (0,19 %)	99,48 % (0,00 %)	65,78 % (2,27 %)
2-1-3	98,14 % (9,92 %)	89,39 % (12,85 %)	95,66 % (9,69 %)	97,32 % (9,84 %)	95,79 % (13,76 %)	92,45 % (9,51 %)	96,08 % (9,72 %)	96,20 % (13,83 %)	82,52 % (9,42 %)
2-2-3	98,44 % (0,00 %)	96,88 % (0,00 %)	91,43 % (0,27 %)	97,91 % (0,05 %)	99,48 % (0,00 %)	82,25 % (0,33 %)	94,25 % (0,26 %)	99,48 % (0,00 %)	66,14 % (0,62 %)
2-3-3	98,44 % (0,00 %)	96,88 % (0,00 %)	91,73 % (0,33 %)	97,92 % (0,00 %)	99,48 % (0,00 %)	82,28 % (0,25 %)	94,29 % (0,09 %)	99,48 % (0,00 %)	66,57 % (0,77 %)
2-4-3	98,44 % (0,00 %)	96,88 % (0,00 %)	92,18 % (0,37 %)	97,92 % (0,00 %)	99,48 % (0,00 %)	82,45 % (0,24 %)	94,27 % (0,00 %)	99,48 % (0,00 %)	67,37 % (0,42 %)
2-5-3	98,44 % (0,00 %)	96,88 % (0,00 %)	92,40 % (0,37 %)	97,92 % (0,00 %)	99,48 % (0,00 %)	82,63 % (0,25 %)	94,27 % (0,00 %)	99,48 % (0,00 %)	67,56 % (0,29 %)
2-5-5-3	97,47 % (9,85 %)	94,94 % (13,63 %)	90,06 % (11,22 %)	96,96 % (9,80 %)	97,49 % (14,00 %)	80,93 % (8,85 %)	93,46 % (9,46 %)	97,50 % (14,00 %)	65,73 % (6,89 %)

MLP on neuroverkon arkitelttuuri. Luokakokohtaiset luokittelutarkkuudet  $C_1$ ,  $C_2$  ja  $C_3$  on ryhmitelty kolmen testijoukon  $A_1$ ,  $A_2$  ja  $A_3$  mukaan. Arvot ovat sadan opetus-/testikerran keskiarvoja. Keskihajonnat ovat sulkeissa.

Taulukko 5.9: Luokkakohittaiset luokittelutarkkuudet 10–250 Hz:n kaistanpäästösuodatuksella.

MLP	$A_1$			$A_2$			$A_3$		
	$C_1$	$C_2$	$C_3$	$C_1$	$C_2$	$C_3$	$C_1$	$C_2$	$C_3$
2-0-3	98,44 % (0,00 %)	96,94 % (0,17 %)	91,74 % (0,55 %)	98,44 % (0,00 %)	99,21 % (0,26 %)	88,57 % (1,39 %)	96,35 % (0,00 %)	99,48 % (0,00 %)	79,08 % (0,81 %)
2-1-3	97,25 % (13,97 %)	90,25 % (9,18 %)	95,25 % (9,06 %)	96,47 % (13,85 %)	94,19 % (9,55 %)	94,34 % (9,65 %)	95,97 % (13,78 %)	96,37 % (9,77 %)	89,88 % (9,22 %)
2-2-3	98,44 % (0,00 %)	96,88 % (0,00 %)	92,71 % (0,07 %)	98,44 % (0,00 %)	98,96 % (0,00 %)	90,10 % (0,00 %)	96,35 % (0,00 %)	99,48 % (0,00 %)	80,21 % (0,11 %)
2-3-3	98,44 % (0,00 %)	96,88 % (0,00 %)	92,94 % (0,28 %)	98,44 % (0,00 %)	98,96 % (0,00 %)	90,10 % (0,00 %)	96,35 % (0,00 %)	99,48 % (0,00 %)	80,51 % (0,36 %)
2-4-3	98,44 % (0,00 %)	96,88 % (0,00 %)	93,17 % (0,21 %)	98,44 % (0,00 %)	98,96 % (0,00 %)	90,10 % (0,00 %)	96,35 % (0,00 %)	99,48 % (0,00 %)	80,88 % (0,38 %)
2-5-3	98,44 % (0,00 %)	96,88 % (0,00 %)	93,21 % (0,26 %)	98,44 % (0,00 %)	98,96 % (0,05 %)	90,08 % (0,26 %)	96,35 % (0,00 %)	99,48 % (0,00 %)	81,09 % (0,37 %)
2-5-5-3	98,45 % (0,16 %)	96,05 % (8,28 %)	91,89 % (9,29 %)	98,45 % (0,16 %)	98,07 % (8,96 %)	89,12 % (9,02 %)	96,39 % (0,37 %)	98,55 % (9,32 %)	79,51 % (8,05 %)

MLP on neuroverkon arkkitehtuuri. Luokkakohittaiset luokittelutarkkuudet  $C_1$ ,  $C_2$  ja  $C_3$  on ryhmitelty kolmen testijoukon  $A_1$ ,  $A_2$  ja  $A_3$  mukaan. Arvot ovat sadan opetus-/testikerran keskiarvoja. Keskihajonnat ovat sulkeissa.

Taulukko 5.10: Luokkakohittaiset luokittelutarkkuudet aaloke-kohinanpoistolla.

MLP	$A_1$			$A_2$			$A_3$		
	$C_1$	$C_2$	$C_3$	$C_1$	$C_2$	$C_3$	$C_1$	$C_2$	$C_3$
2-0-3	98,44 % (0,00 %)	96,88 % (0,00 %)	94,27 % (0,00 %)	98,44 % (0,00 %)	98,96 % (0,00 %)	91,67 % (0,00 %)	96,88 % (0,00 %)	99,48 % (0,00 %)	83,33 % (0,00 %)
2-1-3	98,34 % (9,94 %)	90,46 % (1,19 %)	95,58 % (9,67 %)	97,45 % (9,84 %)	93,82 % (0,93 %)	94,62 % (9,60 %)	97,02 % (9,80 %)	96,09 % (0,82 %)	90,52 % (9,35 %)
2-2-3	98,44 % (0,00 %)	96,35 % (0,00 %)	94,67 % (0,22 %)	98,44 % (0,00 %)	98,46 % (0,14 %)	93,89 % (0,47 %)	96,88 % (0,00 %)	98,98 % (0,10 %)	84,24 % (0,25 %)
2-3-3	98,44 % (0,00 %)	96,35 % (0,00 %)	94,70 % (0,20 %)	98,44 % (0,00 %)	98,60 % (0,24 %)	93,99 % (0,28 %)	96,88 % (0,00 %)	98,96 % (0,00 %)	84,27 % (0,21 %)
2-4-3	98,44 % (0,00 %)	96,35 % (0,00 %)	94,70 % (0,20 %)	98,44 % (0,00 %)	98,76 % (0,26 %)	93,98 % (0,32 %)	96,88 % (0,00 %)	98,96 % (0,00 %)	84,26 % (0,22 %)
2-5-3	98,44 % (0,00 %)	96,35 % (0,00 %)	94,72 % (0,18 %)	98,44 % (0,00 %)	98,84 % (0,22 %)	94,01 % (0,41 %)	96,88 % (0,00 %)	98,97 % (0,07 %)	84,31 % (0,19 %)
2-5-5-3	98,44 % (0,00 %)	96,33 % (0,10 %)	94,72 % (0,18 %)	98,44 % (0,00 %)	98,58 % (0,26 %)	94,08 % (0,45 %)	96,88 % (0,00 %)	98,97 % (0,09 %)	84,66 % (0,61 %)

MLP on neuroverkon arkkitehtuuri. Luokkakohittaiset luokittelutarkkuudet  $C_1$ ,  $C_2$  ja  $C_3$  on ryhmitelty kolmen testijoukon  $A_1$ ,  $A_2$  ja  $A_3$  mukaan. Arvot ovat sadan opetus-/testikerran keskiarvoja. Keskihajonnat ovat sulkeissa.



Tutkielmassa ei vertailla eri tyyppisiä oppimisalgoritmeja. Sen vuoksi taulukoissa 5.3–5.6 esitetyillä iteraatioiden lukumäärillä ei ole kovinkaan suurta merkitystä. Sinänsä algoritmien vertailu olisi vaatinut myös niiden parametrien vaikutuksen huomioimisen. Ilmaisinyksikön toiminnan kannalta keskineliövirheet ja luokittelutarkkuudet ovat tärkeämpiä tunnuslukuja. Iteraatioiden lukumäärien perusteella voidaan kuitenkin havaita, että useimmissa tapauksissa opettamiseen tarvittiin alle kolmesataa iteraatiota.

Opetus- ja validointijoukon keskineliövirheiden perusteella voidaan tehdä alustavia päätelmiä neuroverkon suorituskyvystä. Neuroverkkoja vertailtaessa pienemmästä keskineliövirheestä ei kuitenkaan automaattisesti seurannut parempaa luokittelutarkkuutta. Esimerkiksi aalloke-kohinanpoistoa käytettäessä keskineliövirheet olivat osittain huonompia kuin 10–1000 Hz:n kaistanpäästösuodatuksen tapauksessa. Keskineliövirheet eivät kertoneet koko totuutta, koska aalloke-kohinanpoistolla luokittelutarkkuudet olivat keskimäärin parempia.

Yleisesti keskihajonnat ovat suhteellisen pieniä, kun niitä verrataan keskiarvoihin. Osa keskihajonnoista on käytännössä nollia. Tämän perusteella neuroverkon painokertoimien satunnaisilla alkuarvoilla ei olisi suurta vaikutusta neuroverkkojen oppimiseen. Pienistä keskihajonnoista voidaan myös päätellä, että neuroverkot muotoutuivat lähes samankaltaisiksi eri opetuskerroilla. Siis useimmilla opetuskerroilla saavutettiin lähes yhtä hyvät painokertoimien arvot. Suurimmat keskihajonnat esiintyivät 2–1–3 -verkon tuloksissa. Vaikka kyseinen verkko toimikin parhaiten leveimpien kaistanpäästösuodattimien tapauksissa, viittaavat suuremmat keskihajonnat siihen, että opetuksen alkutilanteella on sen kannalta enemmän merkitystä. Siksi 2–1–3 -verkon opettaminen ja soveltaminen on epävarmempaa kuin muiden verkkojen. Myös suurimman 2–5–5–3 -verkon kohdalla on toisinaan havaittavissa hieman suurempia keskihajontoja.

Luokittelutarkkuuksien parantumista ei ole havaittavissa 2–5–5–3 -verkon kohdalla, mikä viittaa siihen, ettei suuremmilla MLP-neuroverkoilla saavuteta merkittäviä parannuksia luokittelutarkkuuksissa. Ylipäänsä kaikki kaksikerroksisen MLP:n variaatiot olivat lähes yhtä hyviä. Myös yksikerroksinen 2–0–3 -verkko suoriutui tehtävästä hyvin, kun sitä verrataan suurempiin verkkoihin. Tämän perusteella luvun alussa esitelty ilmaisinjärjestelmä voitaisiin rakentaa myös yksikerroksisen neuroverkon ympärille. On kuitenkin syytä huomauttaa, että neuroverkon arkkitehtuurin ohella myös piirteiden erottelulla ja edelleen sopivilla syötearvoilla on huomattava vaikutus neuroverkon suorituskyykyyn.

## Luku 6

### Yhteenveto

Tutkielmassa tarkasteltiin, kuinka neuroverkot soveltuvat lihaksen sähköisen toiminnan muutosten havaitsemiseen. Erityisesti käsiteltiin muutoksia, jotka kertovat tahdonalaisista lihastoiminnoista. Muutosten havaitseminen tapahtui EMG-signaalin perusteella. Tehtävää varten suunniteltiin neuroverkkopohjainen ilmaisinjärjestelmä, jonka toimintaa testattiin kokeellisesti. Tässä vaiheessa keskityttiin monikerrosperceptron-neuroverkkoihin ja järjestelmän toimintaan ei-reaaliaikaisessa ympäristössä. Lisäksi tarkasteltiin sitä, kuinka digitaalisten esikäsittelymenetelmien soveltaminen vaikuttaa neuroverkkojen suoritussykyyn vaihtelevissa kohinaolosuhteissa.

Testeissä vertailtiin seitsemää MLP-neuroverkkoa. Pienimmässä neuroverkossa ei ollut lainkaan piilokerroksia ja se simuloikin yksikerroksista perceptronia. Viisi neuroverkoista oli kaksikerroksisia, joissa oli eri määrä neuroneja yhdessä piilokerroksessa. Lisäksi testattiin kolmikerroksista MLP-neuroverkkoa. Tulosten perusteella piiloneuronien lukumäärällä ei ole kovinkaan suurta merkitystä, sillä kaksikerroksiset neuroverkot toimivat käytännössä yhtä hyvin. Toisen piilokerroksen lisääminen ei parantanut tuloksia, minkä perusteella kaksikerroksinen neuroverkko on riittävä kyseessä olevan tehtävän kannalta. Myös yksikerroksinen neuroverkko osoittautui toimivaksi vaihtoehdoksi, kun sitä vertaillaan suurempiin neuroverkkoihin. Tulosten perusteella MLP-neuroverkot soveltuvat hyvin lihaksen sähköisen toiminnan muutosten havaitsemiseen.

Neuroverkkojen rakenteen ohella tarkasteltiin, kuinka digitaalinen suodatus vaikuttaa neuroverkkojen suoritussykyyn. Testeissä vertailtiin neljää esikäsittelymenetelmää. Kolme niistä olivat kaistanpäästösudattimia ja neljäs menetelmä oli aaloke-kohinanpoisto. Menetelmien vertailu perustui niiden toimintaan vaihtelevissa kohinaolosuhteissa. Kohinaa simuloitiin keinotekoisella valkoisella kohinalla. Kaistanpäästösudattimien tapauksessa tulokset ovat jokseenkin triviaaleja; kaapeammalla päästökaistalla saavutetaan parempi neuroverkkojen suoritussyky kohinaisissa olosuhteissa. Aaloke-kohinanpoistolla saavutettiin parhaimmat luokit-

telutarkkuudet. Se toimi vähintään yhtä hyvin kuin kapein kaistanpäästösuodatin.

Esikäsittelyn lisäksi piirteiden erottelulla on suuri vaikutus neuroverkon ja ilmaisinjärjestelmän suorituskyykyyn. Tutkielmassa neuroverkon syötearvoina käytettiin normalisoituja RMS-arvoja. Testitulosten perusteella ne soveltuivat neuroverkkojen syötearvoiksi tässä yhteydessä hyvin. Jos ilmaisinjärjestelmälle asetetut vaatimukset muuttuvat, on hyvin todennäköistä, että myös neuroverkkoon ja piirteiden erottelijaan on tehtävä muutoksia. Tavoitteena saattaa olla lihasaktiiviteetin muutosten havaitsemisen lisäksi lihaksen toiminnallisen tilan tunnistaminen (ts. määrittää onko lihas levossa vai supistunut). Mikäli normalisoidut RMS-arvot eivät riitä tehtävän vaatimusten saavuttamiseen, on pohdittava myös toisenlaisten piirteiden soveltamista.

Tutkielmassa perehdyttiin MLP-neuroverkkojen soveltamiseen järjestelmässä, joka noudattaa Stauden ja Wolfin [43] geneeristä ilmaisinjärjestelmän mallia. Suurin osa lihasaktiiviteetin muutosten havaitsemiseen käytetyistä menetelmistä noudattaakin kyseistä mallia. Staude ja Wolf vertailevat neljää tällaista menetelmää. Testitulanteesta riippuen kaikilla neljällä menetelmällä saavutettiin noin 95 %:n tarkkuus. Leader *et al.* [26] esittävät parannetun version eräästä algoritmista, jonka avulla saavutettiin keskimäärin 85 %:n tarkkuus. Stylianou *et al.* [44] käyttivät *suurimman uskottavuuden* (maximum likelihood) menetelmää, jolla saavutettiin 80–100 %:n tarkkuus. Sadan prosentin luokittelutarkkuus saavutettiin silloin, kun perusaktiiviteetin taso oli riittävän pieni. Tässä mainittuihin menetelmiin verrattuna neuroverkkopohjainen ilmaisinjärjestelmä toimi erittäin hyvin. Testijoukon kohinasasta sekä sovelletusta esikäsittelymenetelmästä riippuen sen avulla saavutettiin yli 95 %:n luokittelutarkkuus.

Testituloksia on kuitenkin vertailtava harkiten. Tämä johtuu siitä, että niin testiasetelmat kuin analysointimenetelmätkin ovat erilaisia. Toisinaan tulosten vertailu on lähes mahdotonta (katso esim. [23] ja [31]). Lisäksi vertailtavia menetelmiä on testattu reaaliaikaisessa testiympäristössä. Merkittävä jatkotutkimusaihe onkin neuroverkkopohjaisen ilmaisinjärjestelmän toteuttaminen reaaliaikaisena. Silloin on mahdollista toteuttaa testiasetelma, jonka avulla erilaisten menetelmien vertaileminen on helpompaa. Suurin haaste ilmaisinjärjestelmän reaaliaikaistamisessa on todennäköisesti jälkikäsittelijän toteuttaminen. Sen toiminta ei ollut vielä tässä vaiheessa oleellista.

Keinotekoinen lihaksen malli helpottaisi reaaliaikaisen testiasetelman toteuttamista. Erityisesti siinä tapauksessa, jos mallin avulla olisi mahdollista generoida keinotekoisia EMG-signaaleja. Testiasetelmaa on helpompi kontrolloida, kun käytetään keinotekoisia signaaleja. Muutoskohtien tarkkoja esiintymishetkiä on vaikea määrittää aitojen EMG-signaalien perusteella. Keinotekoisien signaalien kohdalla ne tiedetään näytearvon tarkkuudella. Keinotekoisien lihaksen mallin toteuttaminen onkin yksi potentiaalinen jatkotutkimuksen aihe. Malli ei kuitenkaan tee ai-

doista EMG-signaaleista tarpeettomia, koska yleisesti kaikkia järjestelmiä on hyvä testata myös aidolla testidatalla.

Luokkakohhtaisten luokittelutarkkuuksien perusteella lihasaktiiviteetin päättymisen oli tyypillisesti vaikeampi tunnistaa. Tämä johtuu todennäköisesti siitä, että lihasaktiiviteetin päättymisen on hieman hitaampi tapahtuma kuin lihasaktiiviteetin alkaminen. Signaaleissa esiintyvät hitaat muutokset hukkuvatkin helposti kohinan alle. Sopivalla esikäsittelymenetelmällä saattaa olla mahdollista helpottaa tätä ongelmaa. Testitulosten perusteella aalloke-kohinanpoisto toimi paremmin kuin kaistanpäästösuodattimet. Aalloke-kohinanpoiston toimintaan vaikuttavat kuitenkin sen parametrit. Keskeinen parametri on aalloke-muunnoksessa sovellettu äitiaaloke. Eri tyyppisillä äitiaalokeilla on mahdollista saavuttaa erilaisia tuloksia. Eräs potentiaalinen tutkimuskohde onkin aalloke-kohinanpoiston soveltaminen biosignaalien yhteydessä.

Tässä tutkielmassa kohinaolosuhteita simuloitiin keinotekoisien valkoisen kohinan avulla. Jatkossa voidaan vertailla esikäsittelymenetelmien toimintaa myös muiden kohinatyyppien tapauksissa. Keinotekoisen kohinan sijasta voidaan käyttää myös aidosti kohinaisia signaaleja. Kohinaisia signaaleja voidaan mitata esimerkiksi siten, että kohinanvaimennus järjestetään puutteellisesti. Signaalien laatuun voidaan vaikuttaa myös siten, että mittauskohteet jätetään puhdistamatta. Silloin mittauselektrodien väliset impedanssiarvot ovat korkeampia ja signaalit vastaavasti huonompilaatuisia.

Silmänräpäysaktiiviteettia kuvaavia signaaleja ei hyödynnetty tässä tutkielmassa. Aluksi niitä ajateltiin käyttää silmien räpäyksistä syntyvien artefaktien poistamiseen. Koska silmänräpäykset eivät oleellisesti haitanneet muita mittauskanavia, ei silmänräpäysaktiiviteettiä käytetty tähän tarkoitukseen. Toisaalta *corrugator supercilii* -kasvolihaksen aktiiviteetti vuotaa silmänräpäyskanavaan. Sen avulla saatetaan olla mahdollista seurata *corrugator supercilii* -kasvolihaksen toimintaa. Tähän havaintoon ei tutkielmassa perehdytty enempää, mutta tulevaisuudessa sen hyödyntäminen saattaa olla olennaista. Havainto on olennainen erityisesti siinä tapauksessa, jos tavoitteena on mittauksessa käytettävien elektrodien lukumäärän vähentäminen.

Ilmaisinjärjestelmässä käytettiin MLP-neuroverkkoja. Järjestelmän arkkitehtuuri ei kuitenkaan rajoita neuroverkon arkkitehtuuria, kunhan se täyttää vaatimukset syöte- ja tulosneuronien osalta. Jatkossa voidaankin kokeilla myös toisia neuroverkkoarkkitehtuureja. Joitain vaihtoehtoja ovat esimerkiksi radiaalikantafunktioverkot ja niin sanotut viiveverkot. Jälkimmäisissä neuroverkon syötekerroksen neuronit muodostavat sekvenssin, jossa syötearvot siirtyvät jokaisella kierroksella seuraavan syötöneuronin herätearvoksi. Toimintaa voidaan verrata FIR-tyyppisen suodattimen toimintaan. Tavallaan syötearvot muodostavat signaalin, joka kulkee syöte kerrallaan viiveverkon syötekerroksen läpi. Viiveverkon käyt-

tämistä motivoi se, että viiveverkko voisi hyödyntää syötearvojen lisäksi niiden välisiä ajallisia suhteita. Tämänkaltaisesta tiedosta on mahdollisesti hyötyä, kun muistetaan, että EMG-signaali kuvaa ajassa muuttuvaa prosessia. Syötearvojen välisten ajallisten suhteiden huomioiminen saattaakin tehostaa reaaliaikaisen järjestelmän toimintaa.

# Lähdeluettelo

- [1] E. W. Abel, P. C. Zacharia, A. Foster and T. L. Farrow, Neural network analysis of the EMG interference pattern. *Medical Engineering & Physics* **18**, 1 (1996), 12–17.
- [2] Armando B. Barreto, Scott D. Scargle and Malek Adjouadi, A practical EMG-based human-computer interface for users with motor disabilities. *Journal of Rehabilitation Research and Development* **37**, 1 (2000), 53–64.
- [3] Christopher M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1996.
- [4] John T. Cacioppo, Louis G. Tassinary and Gary Berntson, *Handbook of Psychophysiology, 2nd edition*. Cambridge University Press, 2000.
- [5] Arnon Cohen, *Biomedical Signal Processing*. CRC Press Inc., 1986.
- [6] Krista Coleman, Electromyography based human-computer-interface to include movement in elderly persons with movement impairments. In: *Proceedings of the 2001 EC/NSF Workshop on Universal Accessibility of Ubiquitous Computing: Providing for the Elderly*, 75–79.
- [7] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, *Introduction to Algorithms, 2nd edition*. The MIT Press, 2001.
- [8] T. M. Cover, E. Diday, K. S. Fu, A. Rosenfeld, J. C. Simon, T. J. Wagner, J. S. Weszka and J. J. Wolf, *Digital Pattern Recognition*. Springer-Verlag Berlin Heidelberg, 1976.
- [9] Scott Day, Important factors in surface EMG measurement. <http://www.bortec.ca/Images/pdf/EMG%20measurement%20and%20recording.pdf> (4.12.2004), 2002.
- [10] Carlo J. De Luca, Surface electromyography: detection and recording. <http://www.delsys.com/library/papers/SEMGintro.pdf> (4.12.2004), 2002.

- [11] Gianluca De Luca, Fundamental concepts in EMG signal acquisition. <http://www.delsys.com/library/papers/sampling%201-4.pdf> (4.12.2004), 2003.
- [12] Paulo S. R. Diniz, Eduardo A. B. da Silva and Sergio L. Netto, *Digital Signal Processing – System Analysis and Design*. Cambridge University Press, 2002.
- [13] L. Dipietro, A. M. Sabatini and P. Dario, Artificial neural network model of the mapping between electromyographic activation and trajectory patterns in free-arm movements. *Medical & Biological Engineering & Computing* **41** (2003), 124–132.
- [14] Richard O. Duda, Peter E. Hart and David G. Stork, *Pattern Classification, 2nd Edition*. John Wiley & Sons Inc., 2001.
- [15] Torsten Felzer and Bernd Freisleben, HaWCoS: the 'hands-free' wheelchair control system. In: *Proceedings of the Fifth International ACM Conference on Assistive Technologies* (2002), 127–134.
- [16] Alan J. Fridlund and John T. Cacioppo, Guidelines for human electromyographic research. *Psychophysiology* **23**, 5 (1986), 567–589.
- [17] LiMin Fu, *Neural Networks in Computer Intelligence*. McGraw-Hill Inc., 1994.
- [18] Marco Gazzoni, Dario Farina and Roberto Merletti, A new method for the extraction and classification of single motor unit action potentials from surface EMG signals. *Journal of Neuroscience Methods* **136** (2004), 165–177.
- [19] Amara Graps, An introduction to wavelets. <http://www.amara.com/ftpstuff/IEEEwavelet.pdf> (4.12.2004), 1995.
- [20] Richard Wesley Hamming, *Digital Filters, 2nd Edition*. Prentice-Hall Inc., 1983.
- [21] Simon Haykin, *Neural Networks – A Comprehensive Foundation, 2nd Edition*. Prentice Hall International Inc., 1999.
- [22] Robert Hecht-Nielsen, *Neurocomputing*. Addison-Wesley Publishing Company Inc., 1990.
- [23] Paul W. Hodges and Bang H. Bui, A comparison of computer-based methods for the determination of onset of muscle contraction using electromyography. *Electroencephalography and clinical Neurophysiology* **101** (1996), 511–519.
- [24] Emmanuel C. Ifeachor and Barrie W. Jervis, *Digital Signal Processing – A Practical Approach, 2nd Edition*. Pearson Education Limited, 2002.

- [25] Jeff T. Larsen, Catherine J. Norris and John T. Cacioppo, Effects of positive and negative affect on electromyographic activity over zygomaticus major and corrugator supercilii. *Psychophysiology* **40** (2003), 776–785.
- [26] J. K. Leader III, J. R. Boston and C.A. Moore, A data dependent computer algorithm for the detection of muscle activity onset and offset from EMG recordings. *Electroencephalography and clinical Neurophysiology* **109** (1998), 119–123.
- [27] Barney LeVeau and Gunnar B. J. Andersson, Output Forms: Data Analysis and Applications – Interpretation of the Electromyographic Signal. *Selected Topics in Surface Electromyography for Use in the Occupational Setting: Expert Perspectives, U.S. Department of Health and Human Services, NIOSH Pub. No. 91-100, March 1992.*
- [28] Ming Ming Liu, Walter Herzog and Hans H. C. M. Savelberg, Dynamic muscle force prediction from EMG: an artificial neural network approach. *Journal of Electromyography and Kinesiology* **9** (1999), 391–400.
- [29] Carl G. Looney, *Pattern Recognition Using Neural Networks – Theory and Algorithms for Engineers and Scientists*. Oxford University Press, 1997.
- [30] Stéphane Mallat, *A Wavelet Tour of Signal Processing, 2nd Edition*. Academic Press, 2001.
- [31] Silvestro Micera, Angelo M. Sabatini and Paolo Dario, An algorithm for detecting the onset of muscle contraction by EMG signal processing. *Medical Engineering & Physics* **20** (1998), 211–215.
- [32] Tohru Nitta, Solving the XOR problem and the detection of symmetry using a single complex-valued neuron. *Neural Networks* **16** (2003), 1101–1105.
- [33] Alan V. Oppenheim and Ronald W. Schaffer, *Digital Signal Processing*. Prentice-Hall Inc., 1975.
- [34] Timo Partala, Anne Aula and Veikko Surakka, Combined voluntary gaze direction and facial muscle activity as a new pointing technique. In: *Proceedings of the INTERACT 2001*, 100–107.
- [35] N. D. Pah, D. K. Kumar and P. Burton, Adding wavelet decomposition to neural networks for the classification of fatigue SEMG. In: *Proceedings of IEEE Engineering in Medicine and Biology Society Conference* (2001).
- [36] Robi Polikar, The wavelet tutorial. <http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html> (4.12.2004), 1999.



- [37] S. L. Pullman, D. S. Goodin, A. I. Marquinez, S. Tabbal and M. Rubin, Clinical utility of surface EMG. *Neurology* **55**, 2 (2000), 171–77.
- [38] Rangaraj M. Rangayyan, *Biomedical Signal Analysis – A Case-Study Approach*. John Wiley & sons Inc., 2002.
- [39] M. Reischl, R. Mikut, C. Pylatiuk and S. Schulz, Control strategies for hand prosthesis using myoelectric patterns. In: *Proceedings of the 9th Fuzzy Colloquium* (2001), 168–174.
- [40] Steven W. Smith, *The Scientist and Engineer’s Guide to Digital Signal Processing, 2nd Edition*. California Technical Publishing, 1999.
- [41] Kevin Swingler, *Applying Neural Networks – A Practical Guide*. Academic Press, 1996.
- [42] Alcimar Soares, Adriano Andrade, Edgard Lamounier and Renato Carrijo, The development of a virtual myoelectric prosthesis controlled by an EMG pattern recognition system based on neural networks. *Journal of Intelligent Information Systems* **21**, 2 (2003), 127–141.
- [43] G. Staude and W. Wolf, Objective motor response onset detection in surface myoelectric signals. *Medical Engineering & Physics* **21** (1999), 449–467.
- [44] Antonia P. Stylianou, Carl W. Luchies and Michael F. Insana, EMG onset detection using the maximum likelihood method. In: *Proceedings of the 2003 Summer Bioengineering Conference*, 1075–1076.
- [45] Fred J. Taylor, *Digital Filter Design Handbook*. Marcel Dekker Inc., 1983.
- [46] Sergios Theodoridis and Konstantinos Koutroumbas, *Pattern Recognition, 2nd Edition*. Academic Press, 2003.
- [47] David A. Winter, *Biomechanics and Motor Control of Human Movement, 2nd edition*. John Wiley & Sons Inc., 1990.